

UNIVERSITÉ DE NANTES

ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE

SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DES MATÉRIAUX

Année : 2007

N° B.U. :

THÈSE DE DOCTORAT

Diplôme délivré conjointement par l'École Centrale de Nantes et
l'Université de Nantes

Spécialité : AUTOMATIQUE, ROBOTIQUE ET TRAITEMENT DU SIGNAL

Présentée et soutenue publiquement par :

Guillaume GALLOT

13 Décembre 2007

à l'École Centrale de Nantes

TITRE

**Modélisation Dynamique et Commande d'un robot
Anguille**

Jury

Président :	P. BLAZEVIC	Professeur à l'Université de Versailles-Saint-Quentin-en-Yvelines, ISTY
Rapporteurs :	L. JAULIN	Professeur à l'ENSIETA, Université de Brest
	B. JOUVENCEL	Professeur à l'Université de Montpellier, LIRMM
Examineurs	M. ALAMIR	Chargé de Recherche CNRS, université de Grenoble, GIPSA-LAB
	F. BOYER	Maître assistant-(HDR) à l'Ecole des Mines de Nantes, IRCCyN
	W. KHALIL	Professeur à l'Ecole Centrale de Nantes, IRCCyN

Directeur de thèse : **Wisama KHALIL**

Laboratoire : Institut de Recherche en Communications et Cybernétique de Nantes

N° ED : 0366-346

Remerciements

Ce travail a été effectué à l'Institut de Recherche en Communications et en Cybernétique de Nantes, dirigé par M. Jean-François Lafay, au sein de l'équipe de robotique dont Mme Christine Chevallereau est responsable, je les remercie sincèrement tous les deux pour leur accueil.

Je souhaite en premier lieu remercier mon directeur de thèse, M. Wisama Khalil, qui m'a aidé et fait confiance tout au long de cette thèse. C'est grâce à ses conseils et aux connaissances qu'il m'a transmis que j'ai pu au fil de ces trois années évoluer dans mon sujet de thèse.

Je tiens ensuite à remercier M. Pierre Blazevic professeur à l'Université de Versailles-Saint-Quentin-en-Yvelines pour avoir accepté d'être le président de mon jury de thèse. Je remercie également M. Luc Jaulin professeur à l'ENSIETA et M. Bruno Jouvencel professeur à l'université de Montpellier, pour avoir accepté d'être mes rapporteurs, et pour l'intérêt qu'ils ont porté à mon travail. Enfin, je remercie M. Frédéric Boyer Maître assistant à l'Ecole des Mines de Nantes et M. Mazen Alamir chargé de Recherche CNRS à l'université de Grenoble pour avoir accepté de faire partie de mon jury et avec qui j'ai eu l'occasion et le plaisir de travailler durant ces 3 années.

Je remercie tous les doctorants que j'ai côtoyés pendant ces années et avec qui j'ai passé de bons moments. J'ai également eu plaisir à travailler au sein de l'IRCCyN qui est un lieu de convivialité et possédant une agréable ambiance de travail.

Mes derniers remerciements vont à ma famille qui m'a appuyé et soutenu moralement pendant les bons et mauvais moments. Par ce mémoire, je souhaite leur témoigner ma reconnaissance pour la confiance qu'ils ont placée en moi.

Table des matières

Introduction générale	1
1 Etat de l'Art	5
1.1 Les différentes catégories de poissons	5
1.2 Les robots existants et leurs applications	7
1.3 Le robot Anguille de l'IRCCyN	12
1.4 Les enseignements tirés de la biologie	14
1.4.1 Le principe de la nage	14
1.4.2 Modélisation du contact entre le fluide et le poisson	16
1.4.3 Génération de mouvement	17
1.4.4 Les réseaux de neurones	18
1.5 Conclusion	20
2 Modèle Dynamique d'une Anguille à structure sérielle	21
2.1 Introduction	21
2.2 Description de la structure sérielle	23
2.3 Modèle cinématique de la structure	27
2.4 Forme générale des modèles dynamiques	29
2.4.1 Introduction	29
2.4.2 Solution pour le modèle dynamique inverse	30
2.4.3 Solution pour le modèle dynamique direct	30
2.5 Modèle dynamique inverse récursif	31
2.5.1 Introduction	31
2.5.2 Calcul récursif du modèle dynamique inverse de l'anguille	32
2.6 Modèle dynamique direct	36
2.6.1 Introduction	36
2.6.2 Calcul du modèle dynamique direct	36
2.7 Modèle d'interaction fluide-structure	39
2.8 Vérification des algorithmes	44
2.8.1 Calcul du torseur cinétique de l'anguille	45
2.8.2 Générateur de mouvements	47
2.8.3 Validation des calculs	49
2.8.4 Validation croisée des résultats	51
2.9 Exemples de simulations	53
2.9.1 Premier exemple : la nage plane	54
2.9.2 Deuxième exemple : nage 3D en looping	57
2.9.3 Analyse des résultats	60

2.10	Modélisation et résultats en ajoutant des ailerons	64
2.10.1	Modification des algorithmes de Newton-Euler	64
2.10.2	Loi de contrôle pour régler le roulis par les ailerons	66
2.11	Conclusion	69
3	Modélisation d'un module de l'anguille	71
3.1	Introduction	71
3.2	Modèle géométrique inverse du module	73
3.3	Modèles cinématiques du module	79
3.3.1	Définitions	79
3.3.2	Modèle cinématique opérationnel	80
3.3.3	Calcul de la matrice jacobienne inverse	81
3.4	Modèle cinématique inverse du second ordre	84
3.5	Modélisation dynamique inverse du module	85
3.5.1	Définition du modèle dynamique pour les robots à structures fermées	85
3.5.2	Modèle dynamique inverse du module	86
3.5.3	Simplification du modèle dynamique	89
3.6	Calcul de l'espace de travail	93
3.7	Conclusion	100
4	Modèle Dynamique du prototype hybride	101
4.1	Introduction	101
4.2	Modélisation géométrique du prototype	104
4.3	Modèles cinématiques des modules	107
4.3.1	Modèle cinématique inverse	107
4.3.2	Modèle cinématique inverse du second ordre	109
4.4	Modèle dynamique inverse de l'anguille	110
4.4.1	Rappel : Modèle dynamique d'un module	110
4.4.2	Modèle cinématique du robot	111
4.4.3	Calcul du Modèle Dynamique Inverse de l'anguille	112
4.5	Modèle dynamique direct de l'anguille	116
4.5.1	Préliminaires	116
4.5.2	Calcul du Modèle Dynamique Direct de l'anguille	117
4.6	Simulations	119
4.6.1	Générateur de mouvements	119
4.6.2	Modèle d'interaction fluide-structure	121
4.6.3	Validation des algorithmes	123
4.7	Modélisation dynamique du robot en ajoutant des ailerons	126
4.8	Conclusion	128
5	Simulation de la commande de l'anguille	129
5.1	Introduction	129
5.2	Commande de l'anguille	130
5.2.1	Générateur de mouvements	130
5.2.2	Simulation du prototype	138
5.3	Méthode de mesure du cap de l'anguille	142
5.3.1	Présentation du problème	142

5.3.2	Oscillateur de Hopf	144
5.4	Commande en boucle fermée	149
5.4.1	Asservissement en cap	149
5.4.2	Asservissement en altitude	153
5.5	Commande avec perception de l'environnement	158
5.5.1	Description du détecteur d'obstacles	158
5.5.2	Loi de commande proposée	160
5.5.3	Exemple de simulation	161
5.5.4	Simulations avec un ou plusieurs capteurs défectueux	163
5.6	Conclusion	165
Conclusions et perspectives		167
A Expérimentation		177
A.1	Introduction	177
A.2	Présentation du prototype	177
A.2.1	L'architecture informatique	177
A.2.2	Présentation du matériel utilisé	178
A.2.3	La méthodologie de commande adoptée	179
A.3	Commandes bas niveau de chaque module	180
A.3.1	Étude de la boucle de courant	181
A.3.2	Etude de la boucle de position	188
A.3.3	Résultats expérimentaux	191
A.4	Commandes haut niveau de chaque module	193
A.5	Electronique de commande des moteurs	196
A.6	Conclusion	199
B Caractéristiques des moteurs de l'anguille		201
C Calcul du modèle dynamique direct d'un robot manipulateur sériel		205
D Calcul des modèles dynamiques des jambes d'un module		209

Table des figures

1.1	Définitions des différents organes du poisson	5
1.2	Classification des différents modes de nage des poissons (extrait de [Sfakiotakis <i>et al.</i> , 1999])	6
1.3	Le robot BoxyBot de l'EPFL (extrait de [Lachat <i>et al.</i> , 2006])	7
1.4	Le prototype RoboTuna du MIT	8
1.5	Robots Potuna et G9 à nage carangiforme	9
1.6	Le robot REEL2 : Anguille à 5 corps (extrait de [McIsaac et Ostrowski, 1999])	10
1.7	Le robot Lamproie : Robot à base de matériaux à mémoire de forme (extrait de [Ayers <i>et al.</i> , 2000])	10
1.8	Le robot Amphibot II de l'EPFL (extrait de [Ijspeert et Crespi, 2007]) . .	11
1.9	Le robot ACM-R5 de l'institut de technologie de Tokyo	11
1.10	Le prototype de robot anguille avec ses 12 vertèbres	12
1.11	Structure vertébrale du prototype et définition des angles de Cardan	13
1.12	Conception du robot parallèle à partir du fonctionnement musculaire des êtres vivants	13
1.13	Structure mécanique permettant de soutenir la peau en latex	14
1.14	Champ d'écoulement du fluide à l'arrière de l'anguille (extrait de [Tytell et Lauder, 2004])	15
1.15	Représentation d'un neurone	17
1.16	Schéma mathématique du fonctionnement d'un neurone (extrait de [Ekeberg, 1993])	18
1.17	Schéma d'un réseau de neurones pour une vertèbre de l'anguille (extrait de [Ekeberg, 1993])	19
2.1	Structure de la colonne vertébrale d'une anguille et définition des angles de mouvements	23
2.2	Le robot anguille 3D	24
2.3	Forme géométrique de l'anguille	26
2.4	Forces et moments sur le corps j	32
2.5	Décomposition de la vitesse pour le calcul des efforts fluides	39
2.6	Une configuration de l'anguille pendant la simulation dans le vide	49
2.7	Vecteur Quantité de mouvement de l'anguille	50
2.8	Vecteur Moment cinétique de l'anguille	50
2.9	Simulateur de la validation croisée	51
2.10	Différence des accélérations entre le MDI et le MDD	52
2.11	Géométrie de l'anguille	53

2.12	Loi horaire des articulations 3j-2 servant à la propulsion pour $j=4, 7$ et 10 .	54
2.13	Trajectoire de la tête dans le plan $\mathbf{x}_w\text{-}\mathbf{y}_w$ du repère fixe	55
2.14	Vitesse absolue de la tête par rapport au temps	55
2.15	Couples moteurs des articulations 1,16 and 34	56
2.16	Loi horaire des articulations 3j-1 permettant le réglage en altitude	57
2.17	Couples moteurs des articulations 1, 25 et 34	58
2.18	Couples moteurs des articulations 2, 26 et 35	58
2.19	Couples moteurs des articulations 3, 27 et 36	59
2.20	Trajectoire de la tête de l'anguille dans le plan $\mathbf{x}_w\text{-}\mathbf{z}_w$ du repère fixe et configuration de l'anguille toutes les 6 secondes	59
2.21	Vitesse de la tête en fonction de α et pour différentes valeurs de A	60
2.22	Couple moteur maximum en fonction de α et pour différentes valeurs de A	61
2.23	Trajectoire de l'anguille pour différents coefficients du modèle fluide	62
2.24	Angle de Roulis de la tête de l'anguille	63
2.25	Modélisation géométrique du robot sériel avec les ailerons	64
2.26	Comparaison des trajectoires de l'anguille avec et sans ailerons	67
2.27	Angle de roulis de l'anguille durant une nage 3D et avec ou sans ailerons	68
3.1	Muscles extenseurs et fléchisseurs permettant de faire bouger le squelette	71
3.2	Structure et modèle CAO du module parallèle	72
3.3	Schéma de commande	73
3.4	Placement des repères sur le module	74
3.5	Placement des repères pour le tableau des paramètres géométriques	75
3.6	Définition de l'axe de contrainte	82
3.7	Structure arborescente équivalente d'un robot parallèle	85
3.8	Contraintes cinématiques du module	88
3.9	Comparaison des couples pour les 3 articulations motorisées avec le Modèle Dynamique simplifié et Complet	92
3.10	Schéma cinématique équivalent à la mécanique du module	93
3.11	Cône définissant le débattement des rotules Bi	94
3.12	Espace de travail d'un module pour $\theta = 0$	95
3.13	Espace de travail d'un module pour $\theta = \pm 15.3^\circ$	96
3.14	Espace de travail complet du module	97
3.15	Description de la base du module	97
3.16	Intersections de l'ensemble bielle-biellette avec la base	98
3.17	Intersection entre la plate-forme et le moteur de roulis	99
4.1	Model CAO d'un module seul	101
4.2	Positionnement des modules dans le prototype	102
4.3	Assemblage des modules	102
4.4	Placement des repères Σ_k et Σ_{k-1} des modules à l'endroit et à l'envers de l'anguille	104
4.5	Forces et moments exercés sur le module k	113
4.6	Décomposition des vitesses dans le plan transverse d'une tranche de l'anguille	121
4.7	Simulateur pour la validation des algorithmes	124
4.8	Différence des accélérations entre le MDI et le MDD	124
4.9	Comparaison de la vitesse entre le robot série et le robot hybride	125

4.10	Comparaison des couples entre le robot série et le robot hybride	126
4.11	Modélisation géométrique avec les ailerons	126
5.1	Générateur de propulsion à base de CPG	131
5.2	Etude du comportement d'un CPG seul	132
5.3	Simulation du CPG pour différentes valeurs de v et R et $a_i=100$	133
5.4	Etude du comportement de deux CPG Couplés	134
5.5	Simulation de deux CPGs pour différentes valeurs de déphasage $\Delta\phi$	134
5.6	Simulation d'un réseau de CPG pour différentes valeurs des paramètres de contrôle	136
5.7	Simulateur de l'anguille avec un générateur de mouvements à base de CPG	140
5.8	Schéma simulink du générateur de mouvements pour le calcul des consignes de Roulis-Tangage-Lacet	140
5.9	Loi horaire des angles de lacet pour différentes liaisons rotules	141
5.10	Loi horaire des angles de tangage pour différentes liaisons rotules	142
5.11	Données issues des différents capteurs de l'anguille pour estimer le cap . . .	143
5.12	Mise en place d'un filtre pour le calcul de la moyenne	144
5.13	Comportement d'un oscillateur de Hopf	145
5.14	Evolution de ω pour différentes conditions initiales $\omega(0)$ pour la fonction d'apprentissage et ($\mu = 1, \epsilon = 10, \eta = 8$) comme paramètres	146
5.15	Estimation de l'offset ψ_f du signal à apprendre	147
5.16	Application de l'oscillateur proposé pour divers signaux	148
5.17	Stratégie de contrôle pour le cap de l'anguille	149
5.18	Représentation du vecteur de cap $\tilde{\mathbf{x}}_0$ de l'anguille	150
5.19	Application du contrôleur de cap de l'anguille pour de la poursuite de cibles	152
5.20	Application des commandes de cap et d'altitude pour la poursuite de cibles 3D	154
5.21	Angle de roulis de la tête	155
5.22	Consignes calculées par les commandes de cap et d'altitude	155
5.23	Loi de commande pour la poursuite de cibles 3D en utilisant l'angle de tangage des liaisons rotules	157
5.24	Angle de roulis lorsqu'on utilise les angles de tangage des liaisons rotules .	158
5.25	Positionnement des capteurs par rapport à la tête de l'anguille	159
5.26	Simulateur de la commande avec perception de l'environnement	161
5.27	Trajectoire de la tête et plusieurs configurations de l'anguille	162
5.28	Consignes calculées par la loi de commande	163
5.29	Simulations de pannes sur les capteurs de l'anguille	164
5.30	Simulations avec seulement deux capteurs	165
5.31	Principe de l'électrolocation pour le poisson éléphant	166
A.1	Répartition des unités informatiques dans le prototype	178
A.2	Structure du régulateur I-P pour la boucle de courant	181
A.3	Diagramme de Bode pour le régulateur avec et sans filtre	183
A.4	Diagramme de Black pour le régulateur avec et sans filtre	184
A.5	Réponse à un échelon du régulateur avec et sans filtre	185
A.6	Structure du régulateur I-P pour la boucle de courant	186

A.7	Influence de l'anti-windup sur les performances du régulateur en cas de saturation	187
A.8	Caractéristiques d'un moteur à courant continu	187
A.9	Structure du régulateur PID série pour la boucle de position	188
A.10	Boucle de position tenant compte des saturations et des non-linéarités . . .	190
A.11	Test de la boucle de courant	191
A.12	Réponse de la boucle de position pour différentes valeurs de consigne . . .	192
A.13	Consigne de courant calculée par la boucle de position	193
A.14	Automate de contrôle pour les unités informatiques situées dans les vertèbres de l'anguille	194
A.15	Schéma global de fonctionnement de la commande pour un module de l'anguille	195
A.16	Schéma électronique de commande d'un moteur	196
A.17	Alimentation du moteur suivant les phases du signal PWM	197
A.18	Courbe de courant de l'induit du moteur en fonction de la tension appliquée au moteur	197
B.1	Placement des 3 moteurs sur un module	201
D.1	Numérotation des corps pour les paramètres inertiels	209

Liste des tableaux

2.1	Paramètres géométriques de la structure série	25
2.2	Définition de la forme de l'anguille	26
2.3	Paramètres géométriques de la structure arborescente	65
3.1	Tableau des paramètres géométriques	76
3.2	Masses des différents ensembles d'un module du prototype de l'anguille . .	90

Introduction générale

Le développement et l'étude de différents moyens de locomotion est un sujet qui passionne l'être humain depuis des temps immémoriaux. En commençant par l'étude de l'art de se déplacer sur ses deux pieds afin de mieux se comprendre, l'homme s'est peu à peu intéressé aux autres espèces l'entourant afin de mieux comprendre le monde qui l'entoure. C'est ainsi que pendant très longtemps, l'homme a rêvé de pouvoir voler comme les oiseaux dans le ciel, ce qui fût possible au XXème siècle. Aujourd'hui, les challenges ont quelque peu changé : l'homme est capable de se déplacer dans tous les milieux, mais il s'est rendu compte qu'avec toutes les machines qu'il pouvait construire, aucune d'elles ne pouvaient égaler le degré de perfectionnement que la nature a donné aux êtres vivants.

De ce constat, la communauté scientifique s'est tournée vers un domaine au combien passionnant, mais également extrêmement complexe baptisé le Biomimétisme. Cette science, repose sur le principe de la copie des systèmes vivants ou de la conception de systèmes basés sur l'étude de ceux-ci. Ces dernières années, beaucoup de recherches ont été effectuées dans le domaine du Biomimétisme sous-marin, car il constitue d'une part un environnement qu'on connaît peu et d'autre part un domaine où les machines actuelles sont très peu performantes par rapport à ce que peut faire la nature. Ce large engouement pour l'étude des poissons provient d'une étude de Sir James Gray [Gray, 1935] (aussi appelé le paradoxe de Gray) sur les dauphins et qui montre que selon les calculs de l'époque, les muscles du dauphin devraient être 7 fois plus puissants pour lui permettre d'atteindre les vitesses auxquelles il nage. Les différentes études menées ont pu mettre en avant des capacités prodigieuses d'accélération pouvant aller jusqu'à 20 fois la gravité et une extraordinaire manœuvrabilité : par comparaison, un poisson peut faire un virage à 180° sans perdre de vitesse et avec des rayons de courbure de l'ordre d'un dixième de sa longueur [Triantafyllou et Triantafyllou, 1995] alors qu'un bateau, dans les mêmes conditions, perdra la moitié de sa vitesse et aura besoin de 10 fois sa longueur pour faire ce demi-tour.

C'est dans le cadre de ces différentes considérations qu'est né en 2003 le projet ROBEA (ROBotique et Entités Artificielles) Robot Anguille, qui a pour but d'étudier, de concevoir et de réaliser un robot anguille capable de nager en trois dimensions. Ce projet est pluridisciplinaire et regroupe plusieurs laboratoires : Le Muséum National d'Histoire Naturelle

(MNHM), le Laboratoire de Mécanique des Fluides (LMF) de Nantes, le laboratoire d'Automatique de Grenoble de Grenoble (devenu Gipsa-Lab), le Laboratoire d'Informatique, de robotique et de Microélectronique de Montpellier (LIRMM) et le l'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN). Les buts de ce projet sont d'étudier les principes de nages de l'anguille, l'interaction entre le fluide et l'animal et de concevoir un robot dont les caractéristiques se rapprochent de l'être vivant.

Parmi les différents types de poissons qui peuvent exister, l'anguille est considérée comme un optimum en terme de manœuvrabilité pour la nage tridimensionnelle. En effet, c'est son caractère hyper-redondant qui fait la force des poissons à nage anguilliforme. Même si leur anatomie est relativement simple comparée aux autres poissons, le nombre élevé voir quasi-infini de degrés de libertés offre des possibilités de déformations que ne possèdent pas les autres espèces animales. Cette anatomie simple de la colonne vertébrale a permis de concevoir un prototype de robot hyper-redondant dont l'architecture est relativement simple, mais dont les possibilités de mouvements peuvent être extrêmement complexes.

Dans cette thèse, nous allons traiter le problème de la modélisation dynamique du robot anguille en s'inspirant des techniques développées pour la modélisation des robots industriels séries, parallèles ou arborescent et à boucles ouvertes ou fermées. Ainsi, les algorithmes de modélisation développés dans cette thèse utilisent le formalisme des équations de Newton-Euler récursives adaptées au cas des robots où la base n'est pas fixe. Cette thèse aborde également quelques aspects de la commande et du contrôle du robot en simulation et sur le prototype.

Ce mémoire est organisé de la manière suivante :

Le premier chapitre est consacré à l'étude bibliographique des robots aquatiques et biomimétiques, ainsi qu'à la présentation du prototype réalisé dans le cadre de ce projet. Ce chapitre traite également de quelques aspects biologiques liés à la nage et au fonctionnement des réseaux de neurones qui permettent de faire bouger le poisson.

Dans le deuxième chapitre, nous présentons la modélisation dynamique d'un robot anguille sériel hyper-redondant basée sur le formalisme récursif de Newton-Euler. Ce robot dont l'architecture n'a pas été retenue pour la conception du prototype nous a permis d'obtenir les premiers résultats de simulations et de dimensionnement d'un modèle simple du robot. Ainsi, ce chapitre traitera des aspects suivants :

- La modélisation géométrique et cinématique du robot ;
- L'écriture des modèles dynamiques inverse et direct de ce robot ;
- La modélisation simplifiée d'une loi de contact fluide-structure ;

- Un ensemble de simulations permettant d'étudier le comportement du robot dans son milieu aquatique.

Ce chapitre a permis de mettre en avant certains phénomènes liés à la nage 3D des anguilles et qui expliquent certains choix de la nature.

Les travaux de ce chapitre ont permis la publication d'un article dans la conférence *IEEE, International Conference on Robotics and Automation* en 2005 [Khalil *et al.*, 2005], ainsi qu'une publication dans la revue *IEEE Transaction on Systems, Man and Cybernetics* en 2007 [Khalil *et al.*, 2007].

Le chapitre 3 est dédié à l'étude du mécanisme utilisé dans le cadre de la réalisation du prototype. Le robot étant composé d'un ensemble de modules parallèles identiques, ce chapitre traitera donc les aspects de la modélisation géométrique, cinématique et dynamique d'un seul module. Le chapitre abordera également la notion de simplification du modèle dynamique des robots parallèles puis, pour finir, l'étude sur l'espace de travail de ce module.

Le chapitre 4 constitue une capitalisation des chapitres 2 et 3. Le choix d'un robot à structure hybride (robots parallèles assemblés en série) a été choisi pour le prototype pour des raisons de compacité, de résistance et de performance qui sont plus élevées comparé aux robots séries. La modélisation dynamique fait appel aux techniques utilisées dans le chapitre 2 pour l'aspect récursif des calculs, et au chapitre 3 pour l'étude du comportement de chaque structure parallèle. Ainsi, ce chapitre abordera les aspects suivants :

- Adaptation des modèles géométrique et cinématique étudiés dans le chapitre 3 au cas du robot complet ;
- Calcul des modèles dynamiques inverse et direct du prototype ;
- Simulations du robot et comparaison avec les résultats du chapitre 2.

Les travaux menés dans les chapitres 3 et 4 ont donné lieu à la publication d'un article dans la conférence *IEEE, International Conference on Robotics and Automation* en 2007 [Gallot *et al.*, 2007].

Le chapitre 5 aborde la commande en boucle fermée du prototype en simulation. En effet, comme tout système ayant à se déplacer dans un environnement à priori inconnu, celui-ci a besoin de pouvoir adapter ses mouvements au milieu dans lequel il évolue. Pour cela, à partir de différents capteurs, il doit être capable dans un premier temps de percevoir sa situation dans l'environnement et dans un second temps d'en déduire les actions à entreprendre pour atteindre le contexte souhaité. Pour cela, le chapitre abordera les points

suivants :

- L'étude d'un générateur de mouvements basé sur l'étude des réseaux de neurones des poissons à nage anguilliforme ;
- Le traitement des informations provenant des capteurs afin d'en déduire l'état d'évolution du robot ;
- L'étude de quelques lois de commande en boucle fermée de l'anguille.

Ce mémoire se termine par une conclusion générale sur le travail présenté ainsi que plusieurs perspectives dans le cadre de ce projet mais aussi au travers du projet ANR RAAMO (Robot Anguille Autonome pour Milieux Opaques). Cette conclusion est suivie de 4 annexes :

La première annexe aborde les aspects de commande de l'informatique embarquée du prototype. On présentera les différents choix technologiques qui ont été utilisés ainsi que leurs architectures. Seront abordés les problèmes de la commande d'axes pour les asservissements de couple et de position, ainsi que le fonctionnement global de l'anguille.

La seconde annexe donne les caractéristiques techniques des moteurs utilisés dans le projet.

Enfin, les deux dernières présentent des méthodes de calcul permettant de déterminer les modèles dynamiques de l'anguille.

Chapitre 1

Etat de l'Art

L'étude et la conception de robots sous-marins est complexe puisque cela fait appel à de nombreux domaines scientifiques : Mécanique des fluides, modélisation dynamique, conception mécanique,... Si les robots actuellement réalisés sont généralement propulsés à l'aide de systèmes à base d'hélices (robots appelés ROV ou AUV), les évolutions actuelles des robots tendent vers des modes de déplacements inspirés des poissons dont la propulsion est obtenue par des mouvements oscillatoires d'une partie du corps. Aujourd'hui, de multiples robots nageurs existent à travers le monde et nous allons dans ce chapitre faire un tour d'horizon de ces différentes réalisations suivant leurs caractéristiques. De cette vue d'ensemble, nous aurons en notre possession un panel de méthodes qui pourront nous aider dans l'étude et la conception du robot anguille.

1.1 Les différentes catégories de poissons

La multitude d'espèces de poissons vivants dans les océans sont généralement cataloguées suivant les parties du corps qui oscillent afin de générer la nage (figure 1.1).

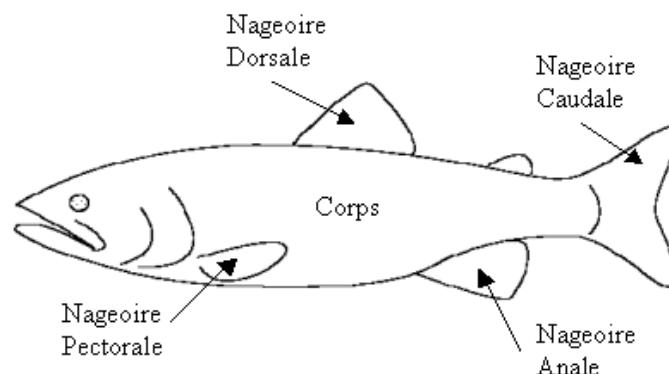


Figure 1.1: – Définitions des différents organes du poisson

Ainsi, on distingue quatre catégories [Sfakiotakis *et al.*, 1999] différentes de poissons que l'on retrouve sur la figure (1.2) :

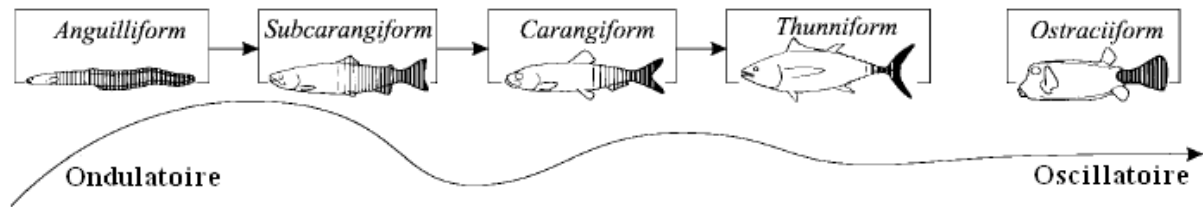


Figure 1.2: – Classification des différents modes de nage des poissons (extrait de [Sfakiotakis *et al.*, 1999])

Cette classification est basée sur la proportion du corps du poisson dont il se sert pour nager. Ainsi on trouvera à l'extrême droite les poissons à nage Ostraciiforme qui n'utilisent que la nageoire caudale, et à l'extrême gauche les poissons à nage anguilliforme qui utilisent la quasi-totalité de leur corps.

Les poissons anguilliformes sont des poissons qui ne peuvent pas atteindre de grandes vitesses, mais qui possèdent de grandes capacités dans les possibilités de manœuvres complexes. De plus, ce type de poisson a la particularité très intéressante de pouvoir nager en marche arrière simplement en inversant le sens du mouvement de l'onde parcourant son corps (les poissons subcarangiformes peuvent également le faire mais plus difficilement). Que ce soit pour les poissons subcarangiformes ou anguilliformes, l'oscillation du corps commence au niveau de la nageoire dorsale. Les deux se distinguent par le fait que les poissons subcarangiformes ont une nageoire dorsale beaucoup plus courte et un peu plus éloignée de la tête.

Les poissons carangiformes sont généralement plus rapides que les poissons anguilliformes mais sont beaucoup moins performants en terme d'accélération et de manœuvrabilité à cause d'une plus grande rigidité de la partie avant du corps. Pour cette catégorie de poissons, l'oscillation du corps se produit à partir de l'arrière de la nageoire dorsale. C'est cette différence qui différencie les poissons subcarangiformes des poissons carangiformes.

Les poissons thunniformes constituent l'optimum en terme de rapidité de nage. Ils sont caractérisés par une large nageoire caudale qui permet de générer une force de poussée importante et leur corps est optimisé afin de réduire les efforts de traînées (ils sont généralement plus larges en hauteur et plus fins en épaisseur). Cependant, ce type de poisson n'est pas adapté pour la nage à faible vitesse et pour effectuer des virages très serrés.

Enfin les poissons ostraciiformes sont un cas particulier puisque l'utilisation de leur nageoire caudale n'est généralement pas considérée comme le mode principal de propulsion (le poisson utilise plutôt ses nageoires pectorales). Les observations montrent que la nageoire caudale est utilisée pour augmenter la vitesse de la nage.

1.2 Les robots existants et leurs applications

A partir des quatre catégories de poissons que nous avons pu voir, nous allons présenter quelques réalisations de robot pour chacune de ces catégories et voir les motivations de ce choix et les résultats obtenus.

- Poissons ostraciiformes :

Pour cette catégorie de poisson, on peut citer le robot BoxyBot de l'EPFL (Ecole Polytechnique Fédérale de Lausanne) [Lachat *et al.*, 2006]. Ce robot est composé de deux nageoires pectorales et d'une queue motorisée (voir figure 1.3) :

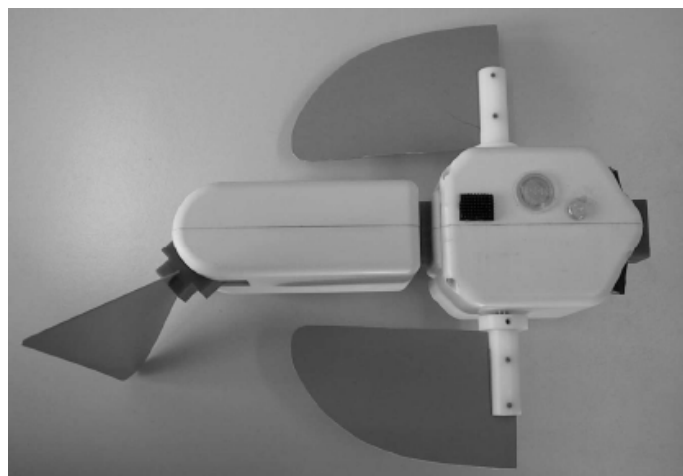


Figure 1.3: – Le robot BoxyBot de l'EPFL (extrait de [Lachat *et al.*, 2006])

Ce robot est conçu dans le but de pouvoir tester des générateurs de locomotion basés sur des systèmes proches des réseaux de neurones. Ce générateur de locomotion est spécifiquement étudié afin de réagir à différents stimuli (nager vers une source lumineuse par exemple) et d'être capable de changer son contexte suivant l'environnement dans lequel il se trouve.

Les résultats sur l'étude de ce robot montrent que ce type poisson possède une très grande manœuvrabilité (notamment grâce aux nageoires pectorales) et permet très aisément de faire de la nage 3D. Cependant, ce genre de robot n'est pas fait pour atteindre des vitesses de déplacement élevées. Le point extrêmement intéressant de cette étude concerne

la conception d'une loi de commande capable de réagir à l'environnement ce qui pourrait s'avérer utile dans le cadre du projet robot anguille.

- Poissons thunniformes :

Dans le domaine des poissons thunniformes, le robot le plus abouti est sans conteste le robot thon Robotuna du MIT [Barrett, 9996] (voir figure 1.4)



Figure 1.4: – Le prototype RoboTuna du MIT

Ce robot a pour but d'étudier les lois de nage permettant d'atteindre de grandes vitesses de déplacement avec la plus faible consommation d'énergie possible. Le mouvement du corps de ce robot thon est obtenu par l'intermédiaire de 8 vertèbres rigides actionnées par 6 moteurs électriques. Ce robot est également le premier à être équipé d'une peau complète et lisse afin d'être le plus proche du poisson qu'il mime.

Les différentes études menées sur ce projet ont permis aux chercheurs de retrouver le paradoxe de Gray et de mettre en évidence le fait que le poisson utilise la création des tourbillons dans son sillage afin d'augmenter l'efficacité de la nage. Ainsi, ces études ont soulevé la question du type de loi de mouvements et du choix des coefficients qui permettent à coup sûr d'atteindre la vitesse maximum.

- Poissons carangiformes ou subcarangiformes :

Dans cette catégorie de poissons, on peut citer le robot Potuna du laboratoire POSTECH [Kim et Youm, 2004] de Corée et le robot G9 de l'université Essex en Angleterre [Hu *et al.*, 2006] (voir figure 1.5) :



a - Robot Potuna (laboratoire POSTECH, Corée) [Kim et Youm, 2004]



b - Robot G9 de l'université Essex (Angleterre) [Hu *et al.*, 2006]

Figure 1.5: – Robots Potuna et G9 à nage carangiforme

Le robot Potuna est un robot dont la queue dispose de deux degrés de liberté et dont les nageoires pectorales sont actionnées par un moteur. Deux types de modélisations ont été utilisées pour étudier le comportement dans l'eau : la théorie des écoulements potentiels (Potential flow theory) en fluide parfait et la modélisation dynamique par la méthode de Kane avec un modèle fluide simplifié de type Morison pour les efforts résistifs avec prise en compte des masses ajoutées. Les simulations réalisées ont pu montrer que l'utilisation de l'une ou l'autre des théories donnent des résultats similaires.

Le robot G9 est un robot qui possède une queue à 4 vertèbres actionnées par 3 moteurs. Le contrôle en altitude de ce robot est assez atypique puisqu'il utilise un système qui permet de modifier la position du centre de gravité. Le cadre d'étude de ce robot est la compréhension et la mise en place de différents modèles de nage qui permettent de réaliser plusieurs catégories de mouvements (virage, plongée,...). Le but de ces travaux est également d'utiliser les informations issues de capteurs afin de décider le type de nage approprié pour le contexte courant.

- Poissons anguilliformes :

Cette catégorie de poissons est celle qui est le plus étudiée dans la littérature. On peut citer le robot REEL 2 de l'université de Harvard [McIsaac et Ostrowski, 1999], le robot lamproie de la Northeastern University [Ayers *et al.*, 2000], le robot lamproie Amphibot II de l'EPFL [Ijspeert et Crespi, 2007] et le robot anguille ACM-R5 du laboratoire Hirose [Yamada *et al.*, 2005].

Le robot REEL2 est un robot dont le but est de pouvoir tester les lois de commande et le problème de la génération de trajectoires, appliqué à un robot 2D nageant en surface.

Les auteurs ont essayé de résoudre les problèmes de la locomotion des systèmes hyper-redundants en combinant les différentes allures de nage d'une anguille biologique (avance, virage, nage latérale) afin d'obtenir les objectifs de commande. Le robot réalisé est constitué de 5 corps en série et reliés par 4 liaisons rotoïdes (dont les axes sont parallèles) motorisées par des servos-moteurs. Bien que d'un aspect assez simple, les auteurs ont pu démontrer leur théorie et obtenir des résultats probants en terme de manœuvrabilité.



Figure 1.6: – Le robot REEL2 : Anguille à 5 corps (extrait de [McIsaac et Ostrowski, 1999])

Le robot lamproie de la Northeastern University [Ayers *et al.*, 2000] est un robot unique en son genre puisqu'il est, de part sa nature, le plus proche du poisson. En effet, son corps n'est pas constitué d'un ensemble de segments actionnés par des moteurs, mais il possède une colonne vertébrale réalisée en matériaux à mémoire de forme. Ainsi, les travaux réalisés ont pour but de générer les différentes lois d'ondulation en utilisant ce type de matériaux et un réseau de neurones connectés aux différents actionneurs. Ce type



Figure 1.7: – Le robot Lamproie : Robot à base de matériaux à mémoire de forme (extrait de [Ayers *et al.*, 2000])

de robot est très intéressant de part son caractère continu, mais les matériaux utilisés actuellement possèdent des rapidités de déformations qui sont trop lentes pour effectuer des nages rapides.

Le robot Amphibot II de l'EPFL [Crespi et Ijspeert, 2006, Ijspeert et Crespi, 2007] est un robot dont le but est de reproduire et d'adapter le fonctionnement de la moelle épinière à différentes classes de robot. Les recherches menées se basent sur l'étude des réseaux de neurones servant à la locomotion et de l'influence de l'environnement extérieur sur ceux-ci. Dans un deuxième temps, les travaux sont axés sur les types et valeurs des signaux

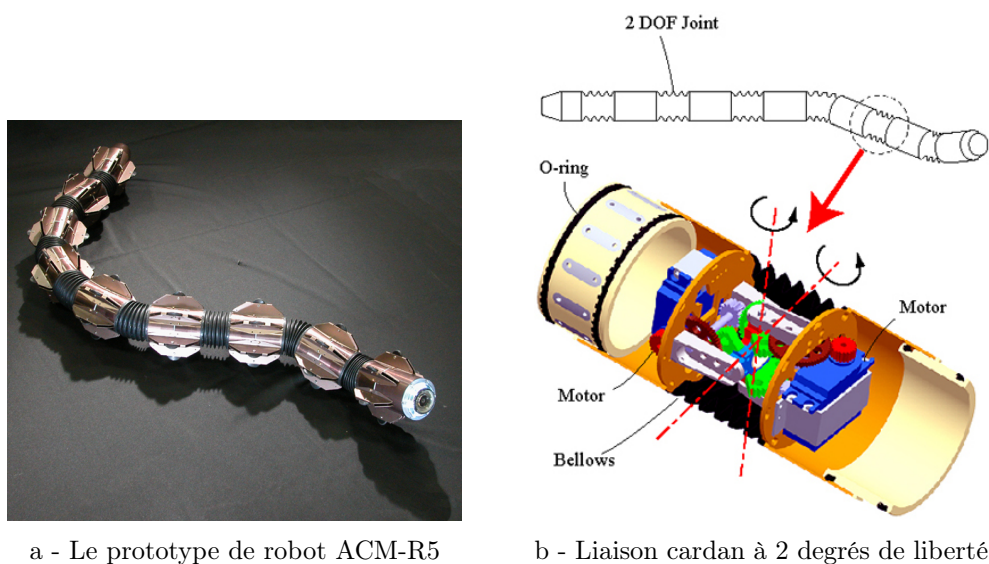
envoyés par le cerveau qui permettent d'obtenir le maximum d'efficacité pendant la nage.



Figure 1.8: – Le robot Amphibot II de l'EPFL (extrait de [Ijspeert et Crespi, 2007])

Sa structure mécanique est constituée de plusieurs modules identiques (le nombre peut varier selon le besoin) et reliés entre eux par une liaison rotoïde (dont les axes sont parallèles) et actionnés par des servomoteurs. Amphibot II est capable de ramper sur le sol, de nager dans l'eau, et d'adapter son mouvement oscillatoire lorsqu'il passe de l'un à l'autre (de manière automatique).

Le robot ACM-R5 du laboratoire Hirose [Yamada *et al.*, 2005] est à notre connaissance le seul (en dehors de celui réalisé dans le cadre du projet robot anguille) à pouvoir nager en 3D (voir figure 1.9 -a).



a - Le prototype de robot ACM-R5

b - Liaison cardan à 2 degrés de liberté

Figure 1.9: – Le robot ACM-R5 de l'institut de technologie de Tokyo

ACM-R5 est capable de se déplacer sur le sol par reptation et dans l'eau par la nage anguilliforme. Il est constitué de neuf segments cylindriques connectés les uns aux autres par une articulation à 2 degrés de liberté de type cardan (figure 1.9-b) et motorisée par 2 servomoteurs. La grande efficacité de nage obtenue par ce robot, tient aux nombreuses

ailettes qui sont réparties tout au long de son corps et qui permettent d'augmenter la surface d'appui sur le fluide (de la même manière que les nageoires sur un poisson). Ce robot est extrêmement performant d'un point de vue réalisation, cependant il n'y a pas eu à l'heure actuelle d'études menées sur la modélisation dynamique ou la conception de lois de commande.

1.3 Le robot Anguille de l'IRCCyN

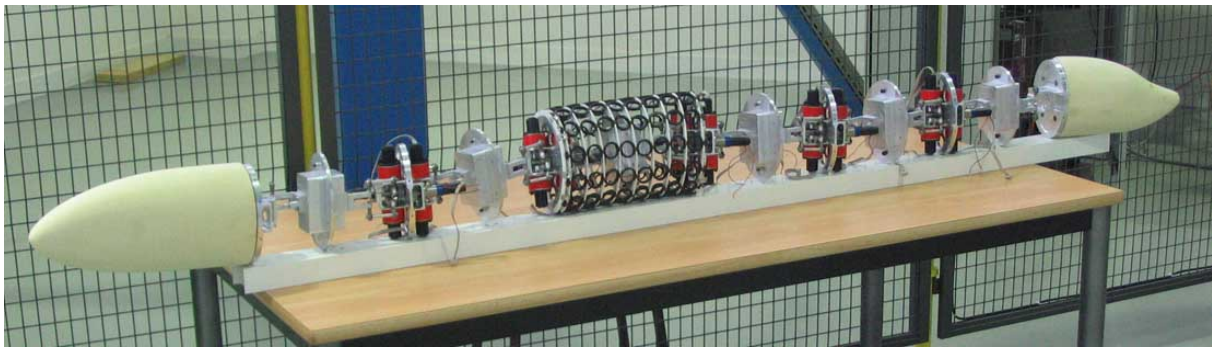


Figure 1.10: – Le prototype de robot anguille avec ses 12 vertèbres

Le prototype de robot présenté figure (1.10) est en cours de réalisation dans le cadre du projet robot anguille [Boyer *et al.*, 2006a]. Ce robot s'inscrit dans un projet dont le but est de développer un engin capable de nager en 3D avec de grandes capacités en terme de manœuvrabilité. Ce projet étant essentiellement axé sur l'aspect bio-mimétique, la conception du robot est fortement inspirée du système vertébral de l'anguille. Même si la technologie actuelle (en terme de miniaturisation) ne nous permet pas de créer un robot avec autant de vertèbres qu'une anguille, nous avons gardé les principes fondamentaux qui permettent d'atteindre l'objectif fixé par ce projet. La caractéristique principale d'une anguille est le caractère hyper-redondant de son squelette, c'est-à-dire qu'il y a beaucoup plus de degrés de liberté internes que de degrés de liberté à piloter. Cette hyper-redondance permet à ces poissons de pouvoir nager ou fouiller dans des endroits très confinés et tortueux sans difficultés.

Le robot réalisé dans ce cadre, est constitué d'un ensemble de modules parallèles connectés en série (ce qu'on appelle robot hybride) qui représentent chacun une articulation de type rotule à 3 degrés de liberté (figure 1.11 -a). Ainsi, la structure mécanique du prototype est constituée de 12 modules, ce qui donne un robot avec 36 degrés de liberté interne. Il faut tout de même noter que, contrairement à notre prototype qui possède 3 degrés de liberté par articulation, l'anguille biologique ne peut avec ses articulations que faire des mouvements autour de l'axe de lacet qui sert à la propulsion (figure 1.11 -b) et de légers

mouvements autour de l'axe de tangage.

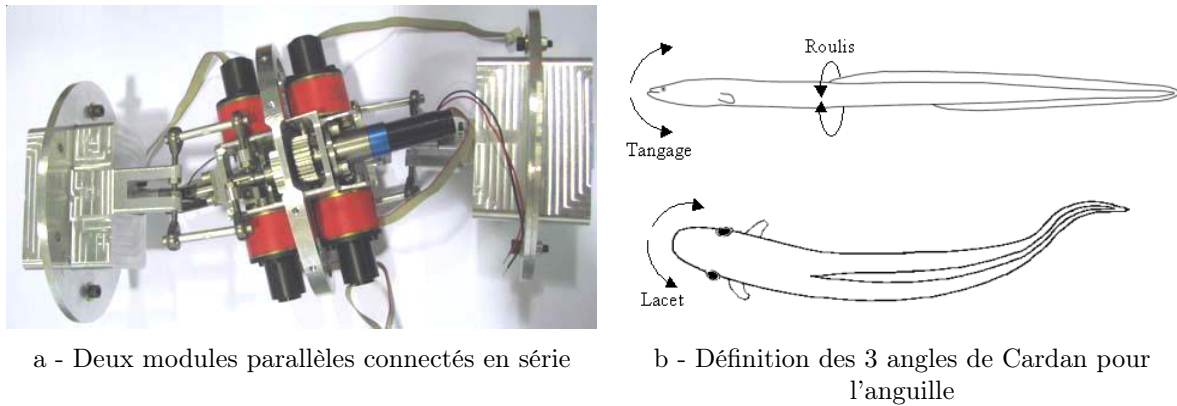


Figure 1.11: – Structure vertébrale du prototype et définition des angles de Cardan

Ce choix d'augmenter les possibilités de mouvements de notre robot par rapport à l'anguille est motivé par la volonté d'améliorer les capacités de déformation afin d'obtenir un prototype plus manœuvrant, mais également de palier au nombre beaucoup plus faible de vertèbres comparé à l'anguille réelle (plus d'une centaine).

L'utilisation de structures parallèles (plutôt que des structures classiques sérielles comme les prototypes que nous avons vus précédemment), nous a paru naturel dans notre approche bio-mimétique. Nous nous sommes basés sur le principe de la paire de muscles antagonistes qu'on retrouve dans la plupart des espèces vivantes (figure 1.12) pour concevoir cette structure. L'avantage principal de cette architecture réside dans la répartition des efforts à fournir puisque, dans notre cas, deux moteurs en parallèles vont être utilisés pour générer le mouvement d'oscillation (on multiplie ainsi la force disponible par deux).

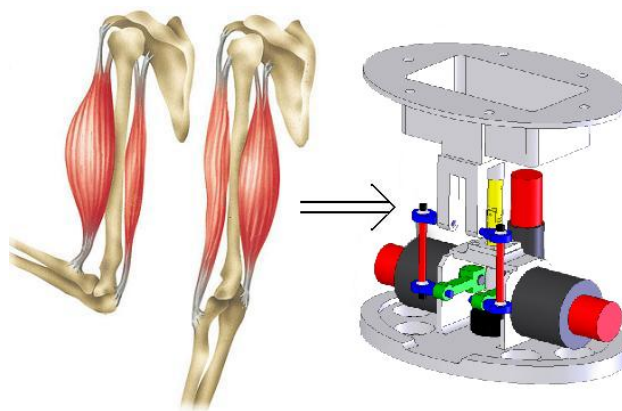


Figure 1.12: – Conception du robot parallèle à partir du fonctionnement musculaire des êtres vivants

Une peau flexible est en cours de montage pour assurer au robot d'une part, une enveloppe continue tout au long de son corps et, d'autre part, une étanchéité pour les différents éléments électriques. Le challenge dans la conception de cette peau est de pouvoir combiner la propriété de continuité tout en minimisant la quantité d'énergie nécessaire pour la déformer. La peau est constituée en deux parties : la première est un ensemble de vertèbres intermédiaires reliées par des anneaux de caoutchouc (figure 1.13) et qui servent de support au deuxième ensemble qui est une enveloppe de latex qui assurera l'étanchéité.



Figure 1.13: – Structure mécanique permettant de soutenir la peau en latex

Toujours dans un aspect bio-mimétique, la tête de notre prototype sera pourvue de 2 nageoires à 1 degré de liberté et qui auront une fonction d'aileron. Elles permettront notamment d'aider à la nage 3D pour la stabilisation des mouvements de roulis qui pourront survenir lors de la nage (problème d'équilibrage, mouvement du à la dynamique,...). En ce qui concerne les aspects informatique et électronique de commande, nous renvoyons le lecteur au chapitre A pour plus de détails.

1.4 Les enseignements tirés de la biologie

Nous avons vu que l'étude des poissons, afin de les imiter, constitue la base du biomimétisme. A partir de ces différentes observations et des différents robots réalisés dans ce domaine, quels sont les enseignements que l'on peut en tirer ?

1.4.1 Le principe de la nage

Pour les robots à nage anguilliforme, on a pu voir que la propulsion est obtenue à partir d'une ondulation générée par la quasi-totalité du corps : cette oscillation est obtenue, in vivo, par la contraction et la relaxation des différents muscles de l'anguille le long de la colonne vertébrale. Mais ce qui fait avancer l'anguille ce n'est pas que le caractère oscillatoire du mouvement, c'est la propagation de cette onde sinusoïdale le long de son corps. Lorsque l'eau arrive en contact avec la peau à l'endroit où commence l'ondulation du

corps, il se crée un tourbillon qui va se propager et s'amplifier tout au long du corps de l'anguille. L'amplification du tourbillon provient de l'augmentation de l'amplitude des oscillations du corps lorsqu'on se dirige vers la queue. Les tourbillons générés de chaque côté du corps de l'anguille, une fois arrivés au niveau de la queue, vont former un sillage propulsif que l'on nomme "Reverse Karman street" [Müller *et al.*, 1997, Tytell et Lauder, 2004] (voir figure 1.14) :

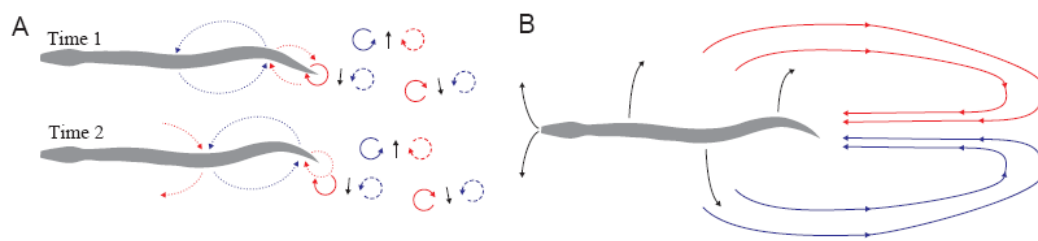


Figure 1.14: – Champ d'écoulement du fluide à l'arrière de l'anguille (extrait de [Tytell et Lauder, 2004])

C'est donc cette combinaison d'une onde de propagation et de l'augmentation de l'amplitude des oscillations lorsqu'on se dirige vers la queue qui permet d'avoir une nage efficace. Des études sur la loi de mouvement utilisée pour la propulsion [Tytell et Lauder, 2004] montrent que l'anguille génère une sinusoïde dont la longueur d'onde est approximativement égale à 60% de la longueur totale de son corps. Ces études ont également montré que l'augmentation de l'amplitude des oscillations pouvait s'écrire sous la forme d'une exponentielle.

Pour pouvoir faire de la nage 3D, on se doute bien que la seule utilisation de l'onde de propulsion que l'on vient de décrire ne suffit pas. Or, les biologistes ont également montré que la partie du corps qui ondule n'est pas capable d'imprimer des déformations dans les autres directions. Pour réaliser les nages 3D, les poissons possèdent plusieurs solutions :

1. Utiliser leurs nageoires pectorales comme gouverne de profondeur ;
2. Déformer la partie non oscillante de leur corps ;
3. Modifier la répartition des masses en jouant sur leur vessie natatoire.

L'utilisation de la vessie natatoire chez le poisson est extrêmement importante puisqu'elle permet d'une part, d'équilibrer le poisson suivant la profondeur de nage (équilibre entre la poussée d'Archimède, la pression de l'eau et la gravité), et d'autre part de régler son assiette. Cependant, compte tenu du peu de place disponible, et des moyens à mettre en

oeuvre pour réaliser cet organe complexe à reproduire et à contrôler, nous avons décidé de ne pas en mettre dans le prototype (il sera simplement lesté).

1.4.2 Modélisation du contact entre le fluide et le poisson

La modélisation des efforts fluides qui sont appliqués au poisson est extrêmement complexe, puisque cela fait appel à la résolution des équations dynamiques de l'eau, de la nage du poisson et des couplages entre les deux. Historiquement ce sont les travaux de Lightill [Lighthill, 1971] qui ont permis de modéliser les écoulements et les champs de pression du fluide autour des poissons. Cependant, ce type de modélisation est très coûteux en temps de calcul et souvent très difficile à résoudre. C'est pourquoi plusieurs autres méthodes ont vu le jour afin de simplifier le problème posé. Parmi ces méthodes, on peut citer la "Panel method" [Galls, 2001] qui consiste à discrétiser la géométrie des poissons en un ensemble de plaques (panneaux). Ensuite, pour chaque intersection entre les panneaux, on modélise le fluide par un ensemble de vortex et de sources créant ainsi un système d'équations qu'il faut résoudre à l'aide de conditions aux limites. Une deuxième technique appelée "Potential flow theory" [Hill, 1998, Mason et Burdick, 1999, Wolfgang *et al.*, 1999] se base sur une hypothèse de fluide irrotationnel et incompressible et permet l'écriture d'équations potentielles traduisant la conservation de la masse et du moment du fluide. Les fonctions de potentiels sont ensuite solutions des équations de Laplace avec conditions aux limites. Enfin, il existe le modèle de Morison [Morison *et al.*, 1950], qui est classiquement utilisé en robotique pour sa simplicité de mise en œuvre et sa rapidité de calcul. Ce modèle est intéressant puisqu'il n'y a pas d'équations de comportement du fluide à résoudre (le fluide est considéré comme idéal) et qu'il dépend simplement de la dynamique du robot. Seulement, ce côté simpliste implique bien souvent (surtout dans les simulations de virage) des résultats qui s'éloignent de la réalité. Ce modèle étant quand même suffisant pour nos besoins, il sera présenté et utilisé dans ce manuscrit. Dans le cadre du projet robot anguille et de la thèse de [Porez, 2007], un nouveau modèle fluide a été proposé qui allie simplicité et performances et qui sera intégré ultérieurement aux travaux de cette thèse. Ce modèle est une fusion de la théorie des "corps minces" pour la mécanique des fluides et de la théorie des "poutres Cosserat" pour la mécanique du solide. Il constitue également une généralisation des travaux de Lightill au cas des robots 3D auto-propulsés.

1.4.3 Génération de mouvement

La génération de l'onde de propulsion chez l'anguille, c'est avant tout une histoire de neurones et d'agencement de ceux-ci. Un neurone est une cellule qui est capable de traiter de l'information et de la diffuser. Il est généralement composé de trois parties (voir figure 1.15) :

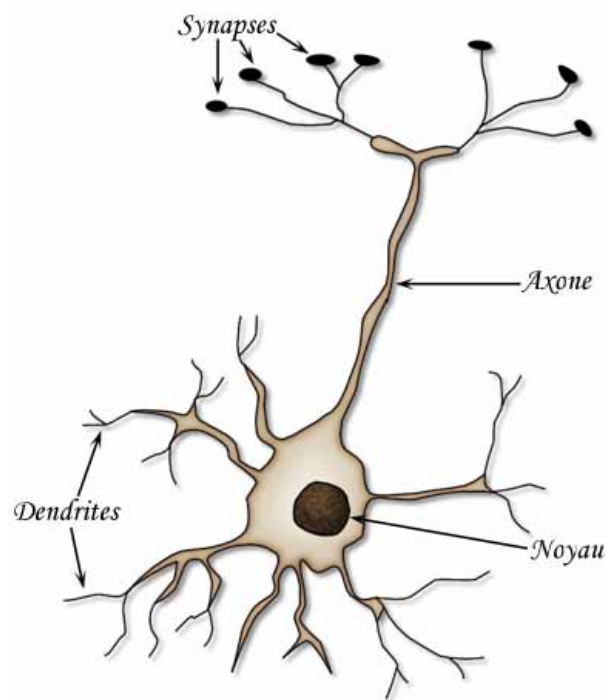


Figure 1.15: – Représentation d'un neurone

- Des dendrites qui collectent les signaux venant d'autres cellules ou de neurones sensoriels ;
- Le soma (ou noyau), qui est le corps du neurone et qui a pour rôle le traitement des informations provenant des dendrites ;
- L'axone, qui permet de transmettre l'information vers les autres neurones.

Le fonctionnement d'un neurone est le suivant : il collecte par l'intermédiaire des dendrites les informations provenant des autres cellules. Ces informations sont ensuite considérées comme excitatrices (signe +) ou inhibitrices (signe -). Le noyau, va ensuite faire une somme pondérée de toutes ces informations puis va transmettre le résultat sous la forme d'un signal électrique en utilisant l'axone.

1.4.4 Les réseaux de neurones

A partir du fonctionnement biologique, Ekeberg a proposé une modélisation sous forme mathématique du comportement d'un neurone [Ekeberg, 1993]. Ce modèle propose un ensemble d'équations différentielles pour les connexions avec les autres neurones et une fonction mathématique du noyau et de la transmission de l'information. Le schéma résultant est présenté sur la figure (1.16) :

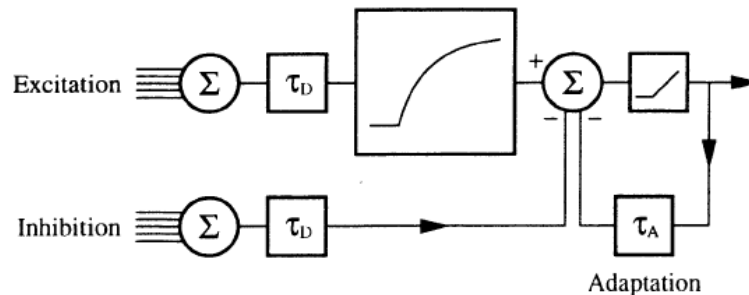


Figure 1.16: – Schéma mathématique du fonctionnement d'un neurone (extrait de [Ekeberg, 1993])

Les blocs τ_d sur la figure (1.16) représentent un terme qui permet de retarder les signaux excitateur et inhibiteur avant le calcul de la somme par le neurone. Ce retard est extrêmement important puisque c'est lui, entre autre, qui permet d'obtenir le phénomène de propagation de la sinusoïde le long du corps de l'anguille. Une fois les signaux retardés, le neurone en fait une somme pondérée et applique une fonction de saturation.

Modèle connexionniste :

Le modèle connexionniste de Ekeberg est une modélisation du réseau de neurone qui permet d'actionner les deux muscles antagonistes pour chaque vertèbre de l'anguille. Le réseau neuronal qu'il présente contient cinq types de neurones :

- MotoNeurones (MN) : il s'agit des cellules qui sont directement connectées aux muscles et qui permettent de commander la contraction par l'envoi d'une impulsion électrique ;
- Exitatory Interneurones (EIN) : ce sont des neurones excitateurs
- Lateral inhibitory InterNeurones (LIN) : ce sont des neurones inhibiteurs latéraux ;
- Contralateral Inhibitory InterNeurones (CCIN) : ce sont des neurones inhibiteurs contralatéraux ;
- Edge Cells (EC) : ce sont des cellules sensorielles.

Le schéma proposé par Ekeberg est représenté sur la figure (1.17) :

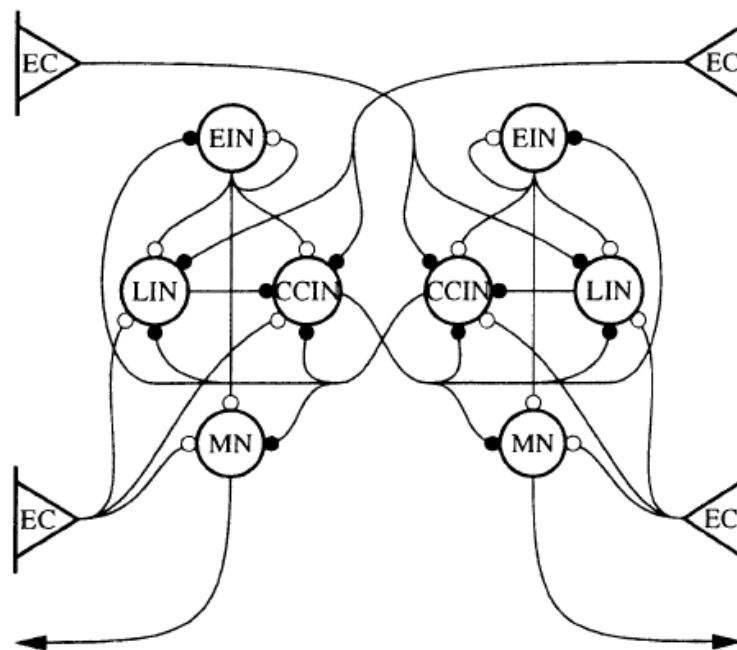


Figure 1.17: – Schéma d'un réseau de neurones pour une vertèbre de l'anguille (extrait de [Ekeberg, 1993])

On peut remarquer que les neurones sont séparés en deux groupes, chacun correspondant à l'activation d'un des muscles antagonistes. Ce qu'il faut retenir de ce schéma c'est qu'il existe un couplage complexe entre ces deux groupes de neurones, ce qui permet de créer une synchronisation entre les muscles de chaque côté de l'anguille. La deuxième chose importante est que ce réseau de neurones est dépendant non seulement du réseau qui le précède (connexions EC en haut de la figure), mais aussi du réseau qui le suit (connexions EC en bas de la figure) : là aussi, cela permet de créer une synchronisation entre les vertèbres consécutives, ce qui permet au final d'obtenir la propagation de la sinusoïde.

La difficulté majeure que l'on rencontre avec ce schéma, est la détermination des poids synaptiques qui existent entre les connexions des différents neurones. Ekeberg a proposé un ensemble de valeurs issue des observations biologiques, mais d'autres chercheurs [Or *et al.*, 2002, Fukunaga *et al.*, 2004] ont proposé des algorithmes génétiques afin de déterminer les valeurs optimales pour la nage.

Les oscillateurs non-linéaires :

Les réseaux de neurones étant complexes dans la modélisation la détermination des poids de connexions entre neurones, de nouveaux modèles mathématiques équivalents ont vu le jour. Ce sont des oscillateurs non-linéaires [Conradt et Varshavskaya, 2003, Inoue *et al.*, 2004, Nakada *et al.*, 2004] dont les signaux de sortie reproduisent ceux générés par l'en-

semble réseau de neurone / muscle. Ces oscillateurs sont caractérisés de la manière suivante :

- un ensemble d'équations différentielles dont la solution est un signal sinusoïdal entretenu ;
- les équations incluent des termes de couplage pour traduire les connexions avec d'autre oscillateur ;
- un certain nombre de paramètres permettant d'ajuster les poids synaptiques et la rapidité de la transmission des informations.

Ce type d'oscillateur est également plus adapté dans le cadre d'expérimentations en robotique. En effet, les réseaux de neurones biologiques génèrent des signaux compatibles avec les muscles : il faut donc proposer un modèle mécanique d'un muscle pour pouvoir l'appliquer sur un robot. Les oscillateurs non linéaires ont l'avantage de produire un mouvement qui est directement assimilable à celui que ferait le muscle.

1.5 Conclusion

Dans ce chapitre sur l'état de l'art, nous avons fait un tour d'horizon sur les principales catégories de poissons que l'on peut trouver, et les différentes réalisations robotiques s'y rapportant. On a pu voir qu'il existe aujourd'hui un panel assez large de robots sous-marins dédiés au biomimétisme. Ceux que nous avons présentés dans ce chapitre couvrent de larges domaines d'études et confèrent ainsi une bonne base de départ pour la réflexion menée dans ce manuscrit. Pour le projet robot anguille, cette réflexion conduit à l'élaboration de 4 grands axes principaux : la conception mécanique, la modélisation dynamique du robot et du fluide l'entourant, la recherche de loi de mouvements permettant d'être le plus efficace possible et enfin la synthèse de systèmes de commande capables de réagir à l'environnement.

Si ces quatre axes principaux ont été largement traités dans la littérature, on trouve cependant très peu de travaux en référence avec la conception et la construction d'une peau censée mimer celle d'un poisson. La solution idéale de la construction d'une peau avec des écailles semble difficilement réalisable aujourd'hui, surtout pour garantir l'étanchéité. Les solutions alternatives qui consistent à utiliser des matériaux de types latex, lycra, etc. sont très intéressantes mais posent encore la question de la consommation en énergie afin de les déformer.

Chapitre 2

Modèle Dynamique d'une Anguille à structure sérielle

Dans ce chapitre, nous présentons le modèle dynamique d'un robot anguille sériel 3D utilisant des algorithmes récursifs basés sur les équations de Newton-Euler. Les modèles dynamiques direct et inverse seront traités. L'algorithme du modèle dynamique inverse permet de calculer l'accélération de la tête ainsi que les couples articulaires en fonction des positions, vitesses et accélérations articulaires ainsi que la vitesse de la tête. Le modèle dynamique direct permet, quant à lui, de calculer l'accélération de la tête et des articulations en fonction des positions, vitesses et couples articulaires ainsi que de la vitesse de la tête. Les algorithmes proposés constituent une généralisation des algorithmes dynamiques récursifs de Newton-Euler pour les robots manipulateurs sériels dont la base est fixe [Luh *et al.*, 1980, Featherstone, 1983]. Les algorithmes sont simples à implémenter et à simuler quel que soit le nombre de degrés de liberté du robot. Un exemple de robot anguille avec 12 articulations rotules sera présenté comme validation. Les efforts fluides ont été pris en compte en utilisant un modèle simple basé sur les équations de Morison [Morison *et al.*, 1950, Susbielles et Bratuch, 1981].

2.1 Introduction

Récemment, beaucoup de projets sont consacrés à l'étude, la réalisation et le contrôle de robots anguilliformes pour leur grand potentiel dans certaines applications sous-marines comme l'inspection océanique ou l'endoscopie industrielle [Ostrowski et Burdick, 1998, McIsaac et Ostrowski, 1999, Mason et Burdick, 2000]. Le travail présenté ici, est réalisé dans le cadre du projet pluridisciplinaire ROBEA "Robot Anguille" supporté par le CNRS. Dans ce projet, la modélisation est basée sur trois niveaux hiérarchiques :

1. un modèle "macro-continu" basé sur la théorie des corps allongés [Lighthill, 1971] et qui constitue le modèle le plus proche du poisson [Boyer *et al.*, 2006b] ;

2. un code Navier-Stokes modélisant les interactions fluide-structure [Leroy et Vissonneau, 2005] et plus exact que les modèles simplifiés utilisés habituellement ;
3. Un modèle de structure articulée rigide qui est destiné à être le plus proche possible du système mécanique pouvant être conçu et réalisé [Khalil *et al.*, 2007].

Ce chapitre est dédié à l'étude du troisième niveau, qui traite un système sériel hyper redondant. Pour le calcul des modèles dynamiques, nous proposons d'utiliser des algorithmes dynamiques récursifs de type Newton-Euler, généralisant ceux utilisés pour les robot manipulateurs à base fixe [Luh *et al.*, 1980, Featherstone, 1983, Khosla, 1986, Khalil et Kleinfinger, 1986, Khalil et Dombre, 2002]. La différence principale entre les deux algorithmes est que l'accélération de la base dans le cas des robots manipulateurs est connue et égale à zéro, tandis que dans le cas d'un robot nageur, la tête est libre et ses accélérations (linéaires et angulaires) doivent être déterminées, que ce soit pour le modèle dynamique direct ou inverse. Les modèles dynamiques proposés sont simples à implémenter et à simuler par des méthodes numériques classiques. En général, le modèle dynamique direct peut être utilisé en simulation, mais le modèle dynamique inverse, qui est généralement utilisé pour la commande, peut également être utilisé en simulation quand on suppose comme négligeable (grâce au système de commande) les erreurs de poursuites des articulations en positions, vitesses et accélérations.

Ce chapitre est organisé de la manière suivante : dans les paragraphes 2.2 et 2.3, nous présentons la modélisation géométrique et cinématique du robot. Ensuite, dans le paragraphe 2.4, nous présentons la forme générale des modèles dynamiques inverse et direct des robots nageurs en utilisant la forme lagrangienne, et nous expliquons pourquoi nous avons préféré utiliser les algorithmes récursifs de type Newton-Euler. Dans les paragraphes 2.5 et 2.6, nous décrivons les différentes étapes pour résoudre le modèle dynamique inverse puis le modèle dynamique direct de notre robot en utilisant les algorithmes de Newton-Euler. Dans le paragraphe 2.7, nous décrivons le modèle d'interaction fluide-structure utilisé pour simuler le robot anguille. Enfin dans les paragraphes 2.8 à 2.10, nous présentons un outil de vérification des algorithmes, des simulations numériques de la nage de l'anguille et pour finir quelques analyses et perspectives sur les résultats obtenus.

2.2 Description de la structure sérielle

L'idée première pour la réalisation de ce robot est de s'inspirer de la structure vertébrale d'une anguille, c'est-à-dire une succession de vertèbres mises bout à bout (en série) et dont les mouvements les uns par rapport aux autres peuvent être assimilés à une liaison pivot (voir figure 2.1) :

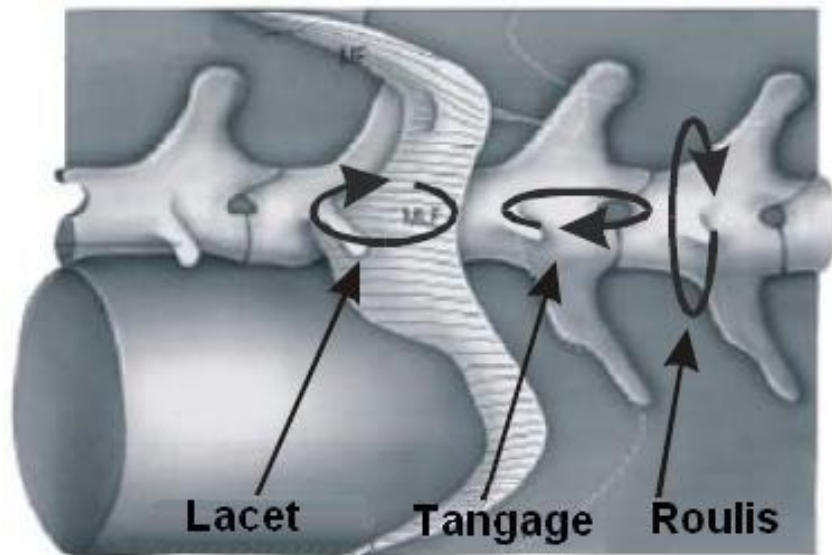


Figure 2.1: – Structure de la colonne vertébrale d'une anguille et définition des angles de mouvements

Pour cette colonne vertébrale, les biologistes ont remarqué que les vertèbres fonctionnent par groupes de 3 : les vertèbres de tangage, de lacet et de roulis. Le regroupement (ou l'assemblage) de ces 3 vertèbres permet d'avoir un comportement mécanique équivalent à une liaison rotule. Malgré tout, cette liaison rotule est fortement contrainte en débattement sur 2 de ses directions : la vertèbre de lacet (qui possède une forme différente) qui est utilisée pour la nage possède un débattement important. Les 2 autres vertèbres peuvent légèrement bouger pour donner de la souplesse à la colonne vertébrale mais possèdent des degrés de débattement très faibles (quelques degrés).

Si une anguille possède plus d'une centaine de vertèbres, le robot que l'on étudie est composé de 13 corps rigides connectés par 12 liaisons rotules. On considère ici que le système étudié est composé d'une séquence de $n+1$ corps connectés par n articulations. Les algorithmes proposés dans ce chapitre peuvent traiter des articulations de type rotoïde ou prismatique. Comme les articulations prismatiques ne sont généralement pas utilisées pour ce type de robots, nous allons limiter la présentation au cas des articulations rotoïdes. Les articulations complexes (cardan ou rotule) peuvent être représentées par une combinaison

équivalente de deux ou trois articulations rotoïdes. Pour notre robot constitué de 12 liaisons rotules, le système complet est par conséquent composé de 36 articulations rotoïdes et de 37 corps.

Nous utilisons les notations de Khalil et Kleinfinger [Khalil et Kleinfinger, 1986, Khalil et Dombre, 2002] pour décrire la structure. Les corps sont numérotés de telle manière que le corps 0 représente la tête du robot et le corps n représente la queue. Pour les 3 articulations qui constituent une liaison rotule k ($k=1,\dots,12$), la numérotation des corps se fait de la manière suivante :

- les corps $3k-2$ et $3k-1$ (représentant les premières et deuxième articulations de chaque rotule), sont des corps virtuels de longueur et de masse nulles, par conséquent aucun effort fluide ne s'appliquent sur eux ;
- les corps $3k$ représentent les vertèbres de l'anguille.

Pour chaque corps j du robot, nous affectons également un repère Σ_j dont l'axe z_j représente l'axe de l'articulation, et l'axe x_j est déterminé par la perpendiculaire commune entre les axes z_j et z_{j+1} . Une partie de la structure de ce robot ainsi que le placement des différents repères sont représentés sur la figure (2.2) :

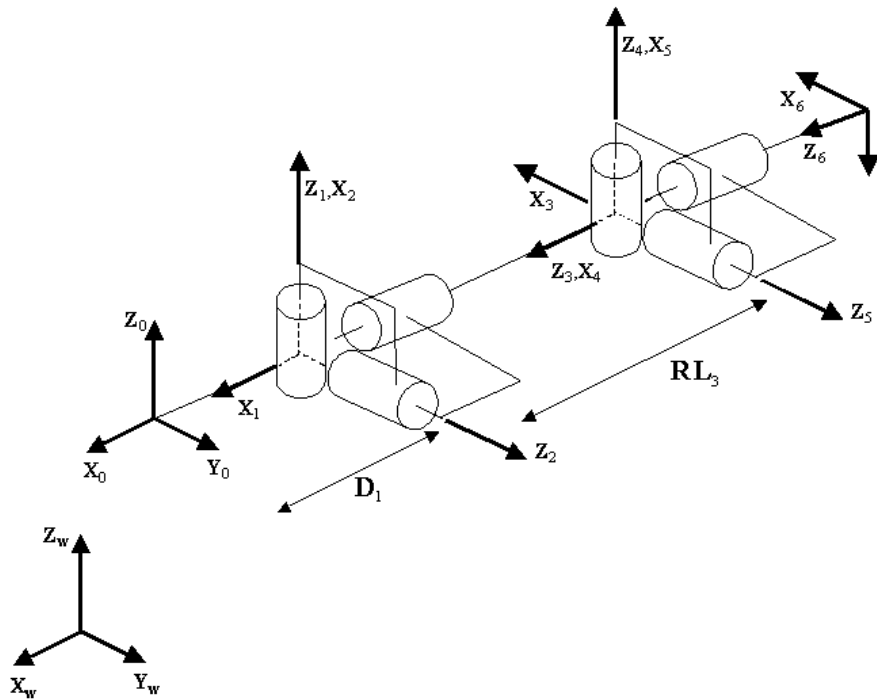


Figure 2.2: – Le robot anguille 3D

Il est à noter que chacune des 3 articulations des liaisons rotules peuvent être utilisées pour générer des mouvements distincts. Ainsi, la première articulation permet d'engendrer un mouvement de lacet, la deuxième un mouvement de tangage et la troisième un

mouvement de roulis.

Pour pouvoir paramétrer complètement ce robot, nous avons besoin de 4 paramètres [Khalil et Kleinfinger, 1986, Khalil et Dombre, 2002] qui permettent d'exprimer la matrice de transformation ${}^{j-1}\mathbf{T}_j$ entre les repères Σ_j et Σ_{j-1} :

- α_j : l'angle entre \mathbf{z}_{j-1} et \mathbf{z}_j autour de \mathbf{x}_{j-1} ,
- d_j : la distance entre \mathbf{z}_{j-1} et \mathbf{z}_j selon \mathbf{x}_{j-1} ,
- θ_j : l'angle entre \mathbf{x}_{j-1} et \mathbf{x}_j autour de \mathbf{z}_j ,
- r_j : la distance entre \mathbf{x}_{j-1} et \mathbf{x}_j selon \mathbf{z}_j .

Les paramètres géométriques de ce robot sont donnés dans le tableau (2.1). Les valeurs d'offsets q_{0j} sont définies de telle manière qu'en prenant $\mathbf{q}_j(\mathbf{0}) = \mathbf{0}$, l'axe \mathbf{x}_0 est aligné avec les axes \mathbf{z}_{3k} des autres corps (avec k le numéro de la liaison rotule, $k=1,\dots,12$).

j	α_j	d_j	q_j	r_j	q_{0j}
1	0	D_1	q_1	0	0
2	$-\pi/2$	0	q_2	0	$\pi/2$
3	$-\pi/2$	0	q_3	RL_3	$\pi/2$
4	$\pi/2$	0	q_4	0	$-\pi/2$
...					
...					
...					
$3n-1$	$-\pi/2$	0	q_{3n-1}	0	$-\pi/2$
$3n$	$-\pi/2$	0	q_{3n}	RL_{3n}	$\pi/2$

Tableau 2.1: – Paramètres géométriques de la structure série

Ce tableau des paramètres géométriques possède la propriété intéressante d'avoir une cyclicité dans l'écriture des lignes, qui correspond à la répétition de l'architecture mécanique pour chaque liaison rotule de l'anguille. Ainsi, pour chaque liaison rotule que l'on souhaite ajouter au robot, il suffit de recopier les lignes 2,3 et 4 du tableau. Enfin, les variables articulaires du robot sont données par :

$$\theta_j = q_j + q_{0j}$$

Les valeurs numériques des paramètres constants du tableau 2.1 sont les suivantes :

$$D_1 = 0,4 \text{ m} \quad , \quad RL_{3k} = 0,14 \text{ m pour } k=1,\dots,12$$

Pour une anguille possédant 12 articulations rotules, à partir des valeurs numériques du tableau des paramètres géométriques, cela donne une longueur de l'anguille égale à 2,08 m.

Description de la forme géométrique de l'anguille

En dehors de la description structurelle de l'anguille, il est également nécessaire de décrire sa forme géométrique globale afin de pouvoir déterminer les paramètres inertiels de chaque corps. La forme de l'anguille adoptée pour la simulations est représentée figure (2.3) :

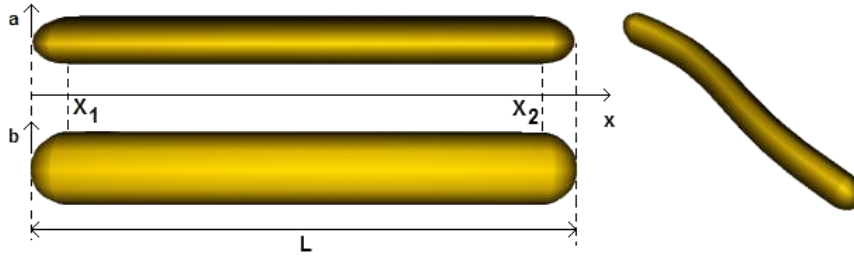


Figure 2.3: – Forme géométrique de l'anguille

La tête de l'anguille (corps 0) est composée d'une demi-sphéroïde et d'un cylindre elliptique. Les corps $3k$, pour $k=1, \dots, 12$ sont composés d'un cylindre elliptique, et le dernier corps, la queue, se termine par une demi-sphéroïde.

La section des corps étant elliptique (dont le petit axe mesure 13 cm et le grand axe mesure 18 cm), on pose $a(x)$ et $b(x)$ comme les demi petit et grand axes de l'ellipse. La paramétrisation complète de l'anguille adoptée pour les simulations est la suivante :

	$a(x)$	$b(x)$
$x \in [0, X_1]$	$a_{max} \sin \left[\arccos \left(\frac{X_1 - x}{X_1} \right) \right]$	$b_{max} \sin \left[\arccos \left(\frac{X_1 - x}{X_1} \right) \right]$
$x \in [X_1, X_2]$	a_{max}	b_{max}
$x \in [X_2, L]$	$a_{max} \sin \left[\arccos \left(\frac{X_2 - x}{L - X_2} \right) \right]$	$b_{max} \sin \left[\arccos \left(\frac{X_2 - x}{L - X_2} \right) \right]$

Tableau 2.2: – Définition de la forme de l'anguille

Avec :

$$X_1=9 \text{ cm}, X_2=1,94 \text{ cm}, L=2.08 \text{ m}, a_{max}=6,5 \text{ cm}, b_{max}=9 \text{ cm}$$

A partir de ces valeurs numériques on peut calculer les paramètres inertiels de chaque corps. Dans ce chapitre, on fait l'hypothèse que le corps de l'anguille est parfaitement

homogène et de densité égale à celle de l'eau ($\rho = 1000 \text{ kg.m}^{-3}$). On obtient alors les paramètres inertiels suivants :

- Le corps de tête (corps 0) :

$$\overrightarrow{O_0 G_0} = \begin{bmatrix} -0.2144 & 0 & 0 \end{bmatrix}^T \quad \mathbf{J}_0 = \begin{bmatrix} 0.0203 & 0 & 0 \\ 0 & 0.4049 & 0 \\ 0 & 0 & 0.3986 \end{bmatrix} \quad M_0 = 6.7999 \text{ kg}$$

- Les corps 3k pour k=1,...,12 :

$$\overrightarrow{O_{3k} G_{3k}} = \begin{bmatrix} 0 & 0 & 0.0700 \end{bmatrix}^T \quad \mathbf{J}_{3k} = \begin{bmatrix} 0.0220 & 0 & 0 \\ 0 & 0.0195 & 0 \\ 0 & 0 & 0.0079 \end{bmatrix} \quad M_{3k} = 2.5730 \text{ kg}$$

- Le dernier corps 3n (la queue) :

$$\overrightarrow{O_{3n} G_{3n}} = \begin{bmatrix} 0 & 0 & 0.0875 \end{bmatrix}^T \quad \mathbf{J}_{3n} = \begin{bmatrix} 0.0179 & 0 & 0 \\ 0 & 0.0166 & 0 \\ 0 & 0 & 0.0042 \end{bmatrix} \quad M_{3n} = 2.5730 \text{ kg}$$

Pour les autres corps (corps 3k-1 et 3k-2 pour k=1,..12), tous les paramètres inertiels sont nuls (ces corps sont fictifs).

2.3 Modèle cinématique de la structure

La modélisation géométrique du paragraphe précédent nous a permis de définir complètement notre robot à partir des 4 paramètres géométriques α_j , d_j , θ_j et r_j [Khalil et Kleinfinger, 1986, Khalil et Dombre, 2002]. A partir de ces paramètres, la matrice (4×4) de transformation entre les repères Σ_{j-1} et Σ_j est exprimée par :

$${}^{j-1}\mathbf{T}_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j S\theta_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

où C^* et S^* représentent respectivement $\cos(*)$ et $\sin(*)$.

Dans la suite, l'exposant en haut à gauche indique le repère de projection. On peut noter que la matrice d'orientation (3×3) notée ${}^{j-1}\mathbf{R}_j$ du repère Σ_j par rapport au repère Σ_{j-1} est la sous-matrice (3×3) en haut à gauche de ${}^{j-1}\mathbf{T}_j$, alors que le vecteur de position ${}^{j-1}\mathbf{P}_j$ définissant l'origine du repère Σ_j par rapport au repère Σ_{j-1} est égal à la sous-matrice (3×1) située en haut à droite.

La matrice de transformation entre le repère fixe lié à la terre Σ_w et le repère lié à la tête de l'anguille Σ_0 est notée par ${}^w\mathbf{T}_0$. Cette matrice est supposée connue à $t=0$, et sera mise à jour en intégrant le vecteur d'accélérations de la tête à chaque pas de temps de l'algorithme. Les vitesses et accélérations cartésiennes de chaque corps (exprimées

dans leurs repères) sont calculées en utilisant les équations récursives suivantes [Khalil et Kleinfinger, 1986, Khalil et Dombre, 2002] :

$${}^j\mathbb{T}_{j-1} = \begin{bmatrix} {}^j\mathbf{R}_{j-1} & -{}^j\mathbf{R}_{j-1} {}^{j-1}\hat{\mathbf{P}}_j \\ \mathbf{0}_{3 \times 3} & {}^j\mathbf{R}_{j-1} \end{bmatrix} \quad (2.2)$$

$${}^j\mathbb{V}_j = {}^j\mathbb{T}_{j-1} {}^{j-1}\mathbb{V}_{j-1} + \dot{q}_j {}^j\mathbf{a}_j \quad (2.3)$$

$${}^j\boldsymbol{\gamma}_j = \begin{bmatrix} {}^j\mathbf{R}_{j-1} \left[{}^{j-1}\boldsymbol{\omega}_{j-1} \times ({}^{j-1}\boldsymbol{\omega}_{j-1} \times {}^{j-1}\mathbf{P}_j) \right] \\ {}^j\boldsymbol{\omega}_{j-1} \times \dot{q}_j {}^j\mathbf{a}_j \end{bmatrix} + \ddot{q}_j {}^j\mathbf{a}_j \quad (2.4)$$

$${}^j\dot{\mathbb{V}}_j = {}^j\mathbb{T}_{j-1} {}^{j-1}\dot{\mathbb{V}}_{j-1} + {}^j\boldsymbol{\gamma}_j \quad (2.5)$$

Avec :

- ${}^j\mathbb{T}_{j-1}$: La matrice (6×6) de transformation entre torseurs ;
- $\hat{\mathbf{P}}$: La matrice (3×3) de pré-produit vectoriel associée au vecteur \mathbf{P} et tel que :

$$\mathbf{P} \times \mathbf{u} = \hat{\mathbf{P}}.\mathbf{u}$$

- \mathbb{V}_j : Le torseur cinématique (6×1) du corps j , défini par :

$$\mathbb{V}_j = \begin{bmatrix} \mathbf{V}_j^T & \boldsymbol{\omega}_j^T \end{bmatrix}^T \quad (2.6)$$

- \mathbf{V}_j : Vitesse de translation de l'origine du repère $\boldsymbol{\Sigma}_j$;
- $\boldsymbol{\omega}_j$: Vitesse angulaire du corps j ;
- ${}^j\mathbf{a}_j$: Matrice de projection (6×1) définie par :

$${}^j\mathbf{a}_j = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ {}^j\mathbf{a}_j \end{bmatrix}$$

- $\mathbf{0}_{m \times n}$: Matrice de zéros de taille $(m \times n)$;
- ${}^j\mathbf{a}_j$: Vecteur unité selon l'axe \mathbf{z}_j , représentant l'axe de l'articulation j .

Ainsi ${}^j\mathbf{a}_j = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

L'initialisation de l'algorithme se fait avec les conditions initiales suivantes :

- Vitesse initiale connue ${}^0\mathbb{V}_0$;
- Position du repère de la tête ${}^w\mathbf{T}_0$ connue.

2.4 Forme générale des modèles dynamiques

2.4.1 Introduction

Le modèle dynamique d'un robot anguille peut être représenté par la relation suivante :

$$\begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \mathbf{\Gamma} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{C} \quad (2.7)$$

Avec :

- $\mathbf{\Gamma}$ ($n \times 1$) : Vecteur des couples articulaires ;
- $\ddot{\mathbf{q}}$ ($n \times 1$) : Vecteur des accélérations articulaires ;
- \mathbf{A} : Matrice d'inertie $(6 + n) \times (6 + n)$ du robot, dont les éléments sont fonction des angles articulaires. Elle peut être partitionnée de la manière suivante :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{A}_{22} \end{bmatrix} \quad (2.8)$$

- \mathbf{A}_{11} : Matrice (6×6) d'inertie du corps composé 0, qui est équivalente à l'inertie de tous les corps du robot, référencés dans le repère Σ_0 (la tête) ;
- \mathbf{A}_{22} : Matrice $(n \times n)$ d'inertie du robot quand la tête est fixe ;
- \mathbf{A}_{12} : Matrice $(6 \times n)$ d'inertie couplée des articulations et de la tête. Il reflète l'effet des accélérations articulaires sur les mouvements de la tête, et l'effet dual des accélérations de la tête sur les mouvements articulaires ;
- \mathbf{C} : Vecteur $(n + 6) \times 1$ représentant les efforts de Coriolis, centrifuge et gravité ainsi que les forces extérieures sur le robot. Ces éléments sont fonction des vitesses de la tête et des articulations ainsi que des efforts extérieurs, en particulier les efforts hydrodynamiques. Ce vecteur peut être partitionné comme ceci :

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \quad (2.9)$$

où :

- \mathbf{C}_1 : Forces de Coriolis, centrifuge, gravité ainsi que les forces extérieures appliquées sur la tête.
- \mathbf{C}_2 : Forces de Coriolis, centrifuge, gravité ainsi que les forces extérieures sur les corps $1, \dots, n$.

2.4.2 Solution pour le modèle dynamique inverse

Dans le cas d'un robot anguille, le modèle dynamique inverse donne les couples articulaires et l'accélération de la tête en fonction de la vitesse de la tête et des positions, vitesses et accélérations articulaires. Ainsi, en utilisant l'équation (2.7), le modèle dynamique inverse est résolu de la manière suivante : premièrement, on utilise la première ligne de l'équation (2.7) pour obtenir l'accélération de la tête :

$${}^0\dot{\mathbf{V}}_0 = -(\mathbf{A}_{11})^{-1}(\mathbf{C}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}}) \quad (2.10)$$

puis la seconde ligne de l'équation (2.2) est utilisée pour calculer les couples articulaires :

$$\mathbf{\Gamma} = \mathbf{A}_{12}^T {}^0\dot{\mathbf{V}}_0 + \mathbf{A}_{22}\ddot{\mathbf{q}} + \mathbf{C}_2 \quad (2.11)$$

2.4.3 Solution pour le modèle dynamique direct

Dans le cas du robot anguille, le modèle dynamique direct donne les accélérations articulaires et les accélérations de la tête en fonction de la vitesse de la tête et des positions, vitesses et couples articulaires. Ainsi, en utilisant l'équation (2.7), le modèle dynamique direct est résolu de la manière suivante : on commence par prendre la première ligne de l'équation (2.7) pour obtenir l'accélération de la tête :

$$\begin{bmatrix} {}^0\dot{\mathbf{V}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} -\mathbf{C}_1 \\ \mathbf{\Gamma} - \mathbf{C}_2 \end{bmatrix} \quad (2.12)$$

Résoudre les problèmes du modèle dynamique inverse et direct en utilisant la forme étendue de Lagrange, comme celle donnée dans l'équation (2.7), est très consommateur en temps de calcul. Ceci est dû au grand nombre de degrés de liberté du système : typiquement n est égal à 36 pour notre robot à 12 liaisons rotules, ce qui donne des expressions très compliquées pour les éléments de la matrice \mathbf{A} et \mathbf{C} . En outre, inverser les matrices d'inertie \mathbf{A}_{11} ou \mathbf{A} pour les modèles dynamiques inverse et direct serait fastidieux même pour la simulation. Les chercheurs qui ont utilisé ce type de méthodes ont limité leur application pour des systèmes planaires 2D [McIsaac et Ostrowski, 1999, Galls, 2001, Kim et Youm, 2004], avec un petit nombre de degrés de liberté internes. Par conséquent, nous proposons ici d'utiliser une méthode récursive, qui est simple à programmer, et dont la complexité peut être grandement simplifiée en utilisant des techniques adaptées de méthodes symboliques itératives [Khosla, 1986, Khalil et Kleinfinger, 1987, Khalil et Creusot, 1997].

2.5 Modèle dynamique inverse récursif

2.5.1 Introduction

Dans ce paragraphe, nous présentons, l'algorithme récursif de Newton-Euler pour calculer le modèle dynamique inverse du robot anguille. Par rapport aux robots manipulateurs, la principale différence réside dans le fait que la tête est libre et que son accélération doit être considérée comme une sortie de l'algorithme. L'algorithme est basé sur les équations cinématiques présentées dans le paragraphe 2.3, donnant les vitesses et accélérations cartésiennes des corps, puis les équations de Newton-Euler donnant la somme de tous les efforts (forces et moments) appliqués sur chaque corps et exprimés dans leurs repères propres.

$${}^j\mathbb{F}_j = {}^j\mathbb{J}_j {}^j\dot{\mathbf{V}}_j + \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times ({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{MS}_j) \\ {}^j\boldsymbol{\omega}_j \times ({}^j\mathbf{J}_j {}^j\boldsymbol{\omega}_j) \end{bmatrix} \quad (2.13)$$

Avec :

- ${}^j\mathbb{F}_j$ torseur de tous les efforts appliqués sur le corps j , donné par :

$$\mathbb{F}_j = \begin{bmatrix} \mathbf{F}_j^T & \mathbf{M}_j^T \end{bmatrix}^T$$

- \mathbf{F}_j somme des forces sur le corps j ;
- \mathbf{M}_j somme des moments des efforts extérieurs sur le corps j par rapport à l'origine O_j ;
- \mathbb{J}_j matrice d'inertie (6×6) du corps j :

$$\mathbb{J}_j = \begin{bmatrix} M_j \mathbf{I}_3 & -\mathbf{MS}_j \\ \mathbf{MS}_j & {}^j\mathbf{J}_j \end{bmatrix} \quad (2.14)$$

- \mathbf{J}_j tenseur d'inertie (3×3) du corps j par rapport au repère Σ_j ,
- \mathbf{I}_3 matrice identité (3×3)
- M_j masse du corps j ,
- \mathbf{MS}_j premier moment d'inertie du corps j par rapport au repère Σ_j .

Par conséquent, l'équation de l'équilibre dynamique de chaque corps est exprimée par (figure 2.4) :

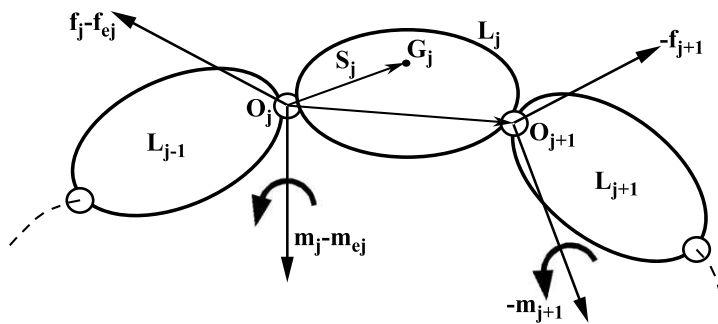


Figure 2.4: – Forces et moments sur le corps j

$${}^j\mathbf{f}_j = {}^j\mathbb{F}_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} + {}^j\mathbf{f}_{ej} \quad (2.15)$$

Avec :

- \mathbf{f}_j le torseur des efforts exercés sur le corps j par le corps j-1 :

$$\mathbf{f}_j = \begin{bmatrix} \mathbf{f}_j^T & \mathbf{m}_j^T \end{bmatrix}^T$$

- \mathbf{f}_{ej} torseur des efforts exercés par le corps j sur l'environnement, et représentant dans notre cas les effets de l'interaction fluide-structure.

2.5.2 Calcul récursif du modèle dynamique inverse de l'anguille

L'algorithme présenté ici peut être considéré comme une généralisation de l'algorithme de Luh, Walker et Paul [Luh *et al.*, 1980] pour les robots manipulateurs. La généralisation concerne le calcul de l'accélération de la tête. Le calcul du modèle dynamique inverse consiste alors dans notre cas, à trois récurrences (une avant, une arrière, puis une avant) au lieu de deux (avant et arrière) pour les robots manipulateurs à base fixe. Pour simplifier l'écriture du modèle dynamique inverse, on suppose que la masse volumique du robot est égale à celle de l'eau de telle manière que le robot puisse naturellement flotter entre deux eaux. Ainsi, sous cette hypothèse, les termes de gravité et de poussée d'Archimède ne seront pas introduits dans les équations qui suivent.

i) Récurrence avant :

Dans cette étape, on calcule la matrice de transformation entre torseurs, la vitesse des corps, et tous les éléments d'accélération et d'efforts des corps qui ne dépendent pas de l'accélération de la tête (${}^0\dot{\mathbf{V}}_0$, ${}^0\dot{\boldsymbol{\omega}}_0$). Ainsi, on calcule pour $j=1, \dots, n$: ${}^j\mathbb{T}_{j-1}$, ${}^j\mathbb{V}_j$ et ${}^j\boldsymbol{\gamma}_j$ en utilisant respectivement (2.2), (2.3) et (2.4). On calcule également ${}^j\boldsymbol{\beta}_j$ qui représente,

dans l'équation (2.15), les termes qui sont indépendants de l'accélération de la tête. A partir des équations (2.13) et (2.15), on note :

$${}^j\beta_j = {}^j\mathbf{f}_{ej} + \begin{bmatrix} {}^j\omega_j \times ({}^j\omega_j \times {}^j\mathbf{MS}_j) \\ {}^j\omega_j \times ({}^j\mathbf{J}_j {}^j\omega_j) \end{bmatrix} \quad (2.16)$$

L'équation (2.15), qui permet d'exprimer l'équilibre des efforts entre corps, peut alors être reformulée en utilisant (2.13) et (2.16) :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j {}^j\dot{\mathbf{V}}_j + {}^j\beta_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} \quad (2.17)$$

ii) *Récurrance arrière* :

Dans cette étape, on détermine l'accélération de la tête à partir des paramètres inertiels du corps composé 0, où le corps composé d'un corps j consiste en la somme des paramètres inertiels de l'ensemble des corps $j, j+1, \dots, n$.

En appliquant les équations de Newton-Euler sur le corps composé j , on obtient en utilisant l'équation (2.17) :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j {}^j\dot{\mathbf{V}}_j + {}^j\beta_j + {}^{j+1}\mathbb{T}_j^T \left({}^{j+1}\mathbb{J}_{j+1} {}^{j+1}\dot{\mathbf{V}}_{j+1} + {}^{j+1}\beta_{j+1} \right) + \dots + {}^n\mathbb{T}_j^T \left({}^n\mathbb{J}_n {}^n\dot{\mathbf{V}}_n + {}^n\beta_n \right) \quad (2.18)$$

En substituant ${}^{j+1}\dot{\mathbf{V}}_{j+1}, \dots, {}^{n-1}\dot{\mathbf{V}}_{n-1}, {}^n\dot{\mathbf{V}}_n$ en fonction de ${}^j\dot{\mathbf{V}}_j$ en utilisant (2.5), on obtient :

$$\begin{aligned} {}^{j+1}\dot{\mathbf{V}}_{j+1} &= {}^{j+1}\mathbb{T}_j {}^j\dot{\mathbf{V}}_j + {}^{j+1}\gamma_{j+1}, \\ {}^{j+2}\dot{\mathbf{V}}_{j+2} &= {}^{j+2}\mathbb{T}_{j+1} {}^{j+1}\dot{\mathbf{V}}_{j+1} + {}^{j+2}\gamma_{j+2} \\ &= {}^{j+2}\mathbb{T}_{j+1} \left({}^{j+1}\mathbb{T}_j {}^j\dot{\mathbf{V}}_j + {}^{j+1}\gamma_{j+1} \right) + {}^{j+2}\gamma_{j+2}, \end{aligned} \quad (2.19)$$

$${}^n\dot{\mathbf{V}}_n = {}^n\mathbb{T}_j {}^j\dot{\mathbf{V}}_j + {}^n\mathbb{T}_{j+2} {}^{j+1}\gamma_{j+1} + \dots + {}^n\gamma_n \quad (2.20)$$

A partir de (2.18), (2.19) on trouve la relation suivante :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j^c {}^j\dot{\mathbf{V}}_j + {}^j\beta_j^c \quad (2.21)$$

avec :

$${}^j\mathbb{J}_j^c = {}^j\mathbb{J}_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbb{J}_{j+1}^c {}^{j+1}\mathbb{T}_j \quad (2.22)$$

$${}^j\beta_j^c = {}^j\beta_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\beta_{j+1}^c + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbb{J}_{j+1}^c {}^{j+1}\gamma_{j+1} \quad (2.23)$$

${}^j\mathbb{J}_j^c$ est la matrice d'inertie spatiale (6×6) du corps composé j . Pour $j = 0$, comme ${}^0\mathbf{f}_0$ est égale à zéro, en utilisant (2.21) on obtient l'équation permettant de calculer l'accélération de la tête de l'anguille :

$${}^0\dot{\mathbf{V}}_0 = - \left({}^0\mathbb{J}_0^c \right)^{-1} {}^0\boldsymbol{\beta}_0^c \quad (2.24)$$

Pour résumer, cette récurrence arrière consiste à initialiser ${}^n\mathbb{J}_n^c = {}^n\mathbb{J}_n$, ${}^n\boldsymbol{\beta}_n^c = {}^n\boldsymbol{\beta}_n$, puis à calculer les équations (2.22) et (2.23) pour $j=n-1, \dots, 0$. Une fois terminé, ${}^0\dot{\mathbf{V}}_0$ est déduit en utilisant l'équation (2.24).

En comparant les équations (2.24) et (2.10), on peut en déduire que le terme \mathbf{A}_{11} est égal à ${}^0\mathbb{J}_0^c$ tandis que ${}^0\boldsymbol{\beta}_0^c$ est égal à $(\mathbf{C}_1 + \mathbf{A}_{12}\ddot{\mathbf{q}})$.

iii) Récurrence avant :

Après avoir calculé ${}^0\dot{\mathbf{V}}_0$ et le torseur dynamique ${}^j\mathbf{f}_j$, les couples moteurs sont obtenus en utilisant les équations (2.5) et (2.21) pour $j=1, \dots, n$:

$${}^j\dot{\mathbf{V}}_j = {}^j\mathbb{T}_{j-1} {}^{j-1}\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\gamma}_j \quad (2.25)$$

$${}^j\mathbf{f}_j = \begin{bmatrix} {}^j\mathbf{f}_j \\ {}^j\mathbf{m}_j \end{bmatrix} = {}^j\mathbb{J}_j^c {}^j\dot{\mathbf{V}}_j + {}^j\boldsymbol{\beta}_j^c \quad (2.26)$$

Le couple moteur est calculé en projetant ${}^j\mathbf{f}_j$ sur l'axe de l'articulation et en tenant compte des frottements et de l'inertie du rotor du moteur par :

$$\boldsymbol{\Gamma}_j = {}^j\mathbf{f}_j^T {}^j\mathbf{a}_j + F_{sj} \text{sign}(\dot{q}_j) + F_{vj} \dot{q}_j + I_{aj} \ddot{q}_j \quad (2.27)$$

Avec :

- F_{sj} paramètre de frottement sec de l'articulation j ;
- F_{vj} paramètre de frottement visqueux de l'articulation j ;
- I_{aj} moment d'inertie du rotor du moteur de l'articulation j ramené au niveau articulaire.

Il est à noter que l'algorithme du modèle dynamique inverse peut être utilisé dans la simulation dynamique du robot anguille en supposant que les positions, vitesses et accélérations articulaires sont égales aux consignes (les erreurs de poursuites sont négligeables). A chaque pas de temps, l'accélération de la tête sera intégrée pour obtenir les vitesses de rotation et de translation qui seront utilisées pour le pas de calcul suivant. L'intégration

de la vitesse de translation permettra de calculer ${}^w\mathbf{P}_0$ représentant la position de l'origine du repère Σ_0 . En ce qui concerne l'orientation, la vitesse angulaire est tout d'abord transformée en vitesse de quaternion selon l'équation suivante [Khalil et Dombre, 2002] :

$$\dot{\mathbf{Q}} = \omega_q \mathbf{Q} \quad (2.28)$$

où :

$$\omega_q = \frac{1}{2 \|\mathbf{Q}\|} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & -\omega_3 & \omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ \omega_3 & -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (2.29)$$

$$\omega = [\omega_1 \quad \omega_2 \quad \omega_3]^T$$

$$\mathbf{Q} = [Q_1 \quad Q_2 \quad Q_3 \quad Q_4]^T$$

L'intégration de l'équation (2.28) permet de calculer le quaternion \mathbf{Q} . Ensuite, la matrice d'orientation ${}^w\mathbf{R}_0$ est calculée en utilisant la relation suivante :

$${}^w\mathbf{R}_0 = \begin{bmatrix} 2(Q_1^2 + Q_2^2) - 1 & 2(Q_2Q_3 - Q_1Q_4) & 2(Q_2Q_4 + Q_1Q_3) \\ 2(Q_2Q_3 + Q_1Q_4) & 2(Q_1^2 + Q_3^2) - 1 & 2(Q_3Q_4 - Q_1Q_2) \\ 2(Q_2Q_4 - Q_1Q_3) & 2(Q_3Q_4 + Q_1Q_2) & 2(Q_1^2 + Q_4^2) - 1 \end{bmatrix} \quad (2.30)$$

où pour une orientation $\mathbf{rot}(\mathbf{u}, \alpha)$, représentant une rotation autour du vecteur unité \mathbf{u} d'un angle α , les composantes du quaternion sont définies comme suit :

$$Q_1 = \cos(\alpha/2), \quad Q_2 = u_x \cdot \sin(\alpha/2)$$

$$Q_3 = u_y \cdot \sin(\alpha/2), \quad Q_4 = u_z \cdot \sin(\alpha/2)$$

$$\text{où } \mathbf{u} = [u_x \quad u_y \quad u_z]^T$$

2.6 Modèle dynamique direct

2.6.1 Introduction

Le modèle dynamique direct d'un robot anguilliforme doit permettre de calculer les accélérations de la tête et les accélérations articulaires en fonction des positions, vitesses et couples articulaires. L'algorithme proposé peut être considéré comme une généralisation de l'algorithme de Featherstone [Featherstone, 1983] qui fut proposé pour les robots manipulateurs sériels à base fixe. Cette fois encore, pour simplifier l'écriture du modèle dynamique direct, on suppose que la masse volumique du robot est égale à celle de l'eau afin de n'introduire ni le terme de gravité, ni le terme de poussée d'Archimède dans les équations qui suivent.

2.6.2 Calcul du modèle dynamique direct

Le modèle dynamique direct est constitué de trois récurrences de même type que celles du modèle dynamique inverse (avant, arrière puis avant) :

i) Récurrence avant :

On calcule les torseurs cinématiques des corps en utilisant les équations (2.2) et (2.3), et les termes d'accélération qui ne dépendent pas des accélérations articulaires et de la tête. A partir de (2.4) et (2.16), on peut alors calculer les équations récursives suivantes pour $j=1, \dots, n$:

$${}^j\zeta_j = \begin{bmatrix} {}^j\mathbf{R}_{j-1} \left[{}^{j-1}\boldsymbol{\omega}_{j-1} \times ({}^{j-1}\boldsymbol{\omega}_{j-1} \times {}^{j-1}\mathbf{P}_j) \right] \\ {}^j\boldsymbol{\omega}_{j-1} \times \dot{q}_j {}^j\mathbf{a}_j \end{bmatrix} \quad (2.31)$$

$${}^j\beta_j = {}^j\mathbf{f}_{ej} + \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times ({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{M}\mathbf{S}_j) \\ {}^j\boldsymbol{\omega}_j \times ({}^j\mathbf{J}_j {}^j\boldsymbol{\omega}_j) \end{bmatrix} \quad (2.32)$$

ii) Récurrence arrière :

Dans cette étape, on détermine l'accélération de la tête à partir des paramètres inertiels du corps composé 0 (même principe que pour le calcul du Modèle Dynamique Inverse). En utilisant les équations (2.13), (2.15) et (2.16) on peut réécrire l'équation d'équilibre du corps j :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j {}^j\dot{\mathbf{V}}_j + {}^j\beta_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} \quad (2.33)$$

Avec :

$${}^j\dot{\mathbf{V}}_j = {}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j \ddot{q}_j + {}^j\zeta_j \quad (2.34)$$

Alors :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j\ddot{\mathbf{q}}_j + {}^j\zeta_j \right) + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} + {}^j\boldsymbol{\beta}_j \quad (2.35)$$

Cependant, dans le cas du modèle direct, les accélérations articulaires ne sont pas connues. Pour pouvoir se passer de ce terme dans les équations, la méthode consiste à exprimer $\ddot{\mathbf{q}}_j$ et ${}^j\mathbf{f}_j$ en fonction de ${}^{j-1}\dot{\mathbf{V}}_{j-1}$. En remarquant que pour le corps n ${}^{n+1}\mathbf{f}_{n+1} = 0$ (le corps n représente l'organe terminal et la queue de l'anguille), l'équation (2.35) donne pour $j=n$:

$${}^n\mathbf{f}_n = {}^n\mathbb{J}_n \left({}^n\dot{\mathbf{V}}_{n-1} + {}^n\mathbf{a}_n\ddot{\mathbf{q}}_n + {}^n\zeta_n \right) + {}^n\boldsymbol{\beta}_n \quad (2.36)$$

En généralisant la méthode pour les autres corps ($j=n-1, \dots, 1$), on peut alors déterminer les équations récursives pour $j=1, \dots, n$ (le détail permettant d'obtenir ces équations se trouve en annexe C) :

$${}^j\mathbf{f}_j = {}^j\mathbb{K}_j {}^j\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\alpha}_j \quad (2.37)$$

$$\ddot{\mathbf{q}}_j = \mathbf{H}_j^{-1} \left(-{}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\zeta_j \right) + \boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^* \right) \quad (2.38)$$

avec :

$$\mathbf{H}_j = {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j + \mathbf{I}_{a_j} \quad (2.39)$$

$${}^j\mathbb{K}_j = {}^j\mathbb{J}_j^* - {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j \mathbf{H}_j^{-1} {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \quad (2.40)$$

$$\boldsymbol{\tau}_j = \boldsymbol{\Gamma}_j - \mathbf{F}_{sj} \text{sign}(\dot{\mathbf{q}}_j) - \mathbf{F}_{vj} \dot{\mathbf{q}}_j \quad (2.41)$$

$${}^j\boldsymbol{\alpha}_j = {}^j\mathbb{K}_j {}^j\zeta_j + {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j \mathbf{H}_j^{-1} (\boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^*) + {}^j\boldsymbol{\beta}_j^* \quad (2.42)$$

$${}^{j-1}\mathbb{J}_{j-1}^* = {}^{j-1}\mathbb{J}_{j-1} + {}^j\mathbb{T}_{j-1}^T {}^j\mathbb{K}_j {}^j\mathbb{T}_{j-1} \quad (2.43)$$

$${}^{j-1}\boldsymbol{\beta}_{j-1}^* = {}^{j-1}\boldsymbol{\beta}_{j-1} + {}^j\mathbb{T}_{j-1}^T {}^j\boldsymbol{\alpha}_j \quad (2.44)$$

On note que pour la tête ($j=0$), les équations (2.39)...(2.42) ne sont pas valides parce qu'il n'existe pas de moteur. En utilisant l'équation (2.33) et en remarquant que ${}^0\mathbf{f}_0$ est égal à zéro, on obtient :

$$0 = {}^0\mathbb{J}_0 {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0 + {}^1\mathbb{T}_0^T {}^1\mathbf{f}_1 \quad (2.45)$$

A partir de l'équation (2.37) pour $j=1$, l'équation (2.45) devient alors :

$$0 = {}^0\mathbb{J}_0 {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0 + {}^1\mathbb{T}_0^T \left({}^1\mathbb{K}_1 {}^1\mathbb{T}_0 {}^0\dot{\mathbf{V}}_0 + {}^1\boldsymbol{\alpha}_1 \right) \quad (2.46)$$

$$0 = {}^0\mathbb{J}_0^* {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0^* \quad (2.47)$$

La résolution de cette équation donne :

$${}^0\dot{\mathbf{V}}_0 = - \left({}^0\mathbb{J}_0^* \right)^{-1} {}^0\boldsymbol{\beta}_0^* \quad (2.48)$$

Pour résumer, on commence par initialiser, ${}^n\mathbb{J}_n^* = {}^n\mathbb{J}_n$, ${}^n\boldsymbol{\beta}_n^* = {}^n\boldsymbol{\beta}_n$ puis on calcule pour $j=n, \dots, 1$ les équations (2.39) à (2.44). L'accélération de la tête de l'anguille ${}^0\dot{\mathbf{V}}_0$ est calculée en utilisant l'équation (2.48).

Remarque :

On note que ${}^0\boldsymbol{\beta}_0^*$ est exprimé en fonction de $\boldsymbol{\tau}$, alors que ${}^0\boldsymbol{\beta}_0^c$ de l'équation (2.24) est exprimé en fonction de $\ddot{\mathbf{q}}$.

iii) Récurrence avant :

Après avoir calculé l'accélération de la tête ${}^0\dot{\mathbf{V}}_0$, le torseur ${}^j\mathbf{f}_j$ (si nécessaire) ainsi que les accélérations articulaires $\ddot{\mathbf{q}}_j$ sont calculés pour $j=1, \dots, n$ en utilisant les équations (2.37) et (2.38) :

$$\ddot{\mathbf{q}}_j = \mathbf{H}_j^{-1} \left[- {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\zeta}_j \right) + \boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^* \right] \quad (2.49)$$

$${}^j\mathbf{f}_j = {}^j\mathbb{K}_j {}^j\mathbb{T}_{j-1} {}^{j-1}\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\alpha}_j \quad (2.50)$$

où :

$${}^j\dot{\mathbf{V}}_j = {}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j \ddot{\mathbf{q}}_j + {}^j\boldsymbol{\zeta}_j \quad (2.51)$$

A noter que l'algorithme du modèle dynamique direct peut être utilisé dans la simulation dynamique du robot anguille en supposant que les positions, vitesses et couples articulaires sont connus. Comme pour la simulation du modèle dynamique inverse, l'accélération de la tête sera intégrée à chaque pas de temps pour obtenir les vitesses de rotation et de translation qui seront utilisées pour le pas de calcul suivant. La position du robot sera obtenue par intégration de la vitesse de translation tandis que la matrice d'orientation sera obtenue par l'intégration du quaternion de vitesse (équations 2.28 à 2.30).

2.7 Modèle d'interaction fluide-structure

Pour simuler la nage d'un robot de type anguille, nous avons adopté un modèle simple de mécanique des fluides. On suppose ici, que les forces exercées par le fluide sur un corps donné du robot sont uniquement dues au mouvement de ce corps. De plus, on présume que les corps sont de forme elliptique dont l'assemblage en série donne un cylindre à forme variable, dont l'une des ses trois dimensions caractéristiques (la dimension axiale) est plus grande (avec un facteur supérieur à 10) que les deux autres (les dimensions transverses). Ainsi, on peut utiliser la théorie des corps minces dans le cadre de la mécanique des fluides [Katz et Plotkin, 2002]. Fondé sur cette hypothèse, le déplacement 3D du fluide autour du corps peut être remplacé par une découpe continue de l'anguille en tranches d'épaisseurs infinitésimales, sur lesquelles le fluide se comporte comme un écoulement plan transverse. Par conséquent, les efforts fluides peuvent être modélisés comme un champ d'efforts appliqués sur chaque tranche de ce corps, et qui dépendra uniquement des mouvements transverses de celui-ci. Ceci correspond à l'approche dite "théorie des bandes" classiquement utilisée en hydrodynamique navale [Lewis, 1989]. A noter tout de même qu'un tel modèle néglige les perturbations dues aux intersections des corps au niveau des articulations.

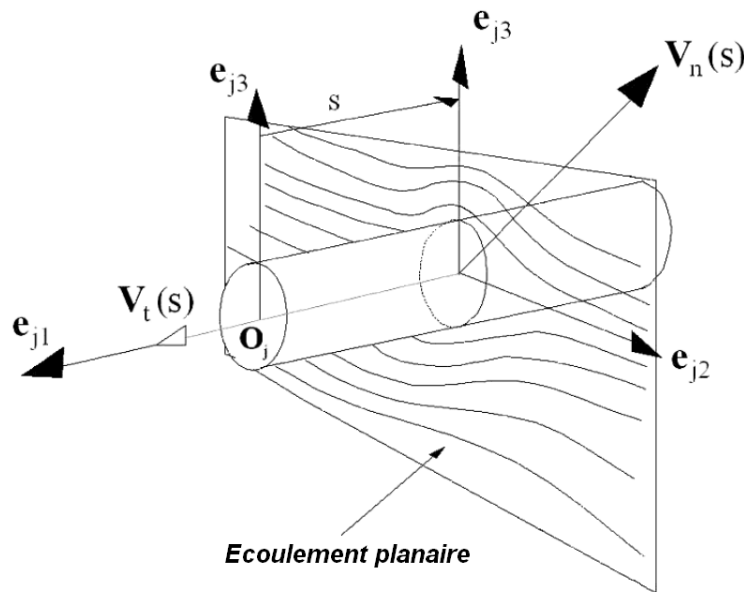


Figure 2.5: – Décomposition de la vitesse pour le calcul des efforts fluides

Le repère (e_{j1}, e_{j2}, e_{j3}) représenté sur la figure (2.5) est défini de la manière suivante :

- pour la tête : $e_{01} = \mathbf{x}_0$, $e_{02} = \mathbf{y}_0$, $e_{03} = \mathbf{z}_0$,
- pour les corps $3j$: $e_{3j1} = \mathbf{z}_{3j}$, $e_{3j2} = \mathbf{x}_{3j}$, $e_{3j3} = \mathbf{y}_{3j}$

On pose $\mathbf{V}_j(s)$ comme la vitesse d'une section du corps j située à une distance s du point O_j selon l'axe du corps (voir figure 2.5). Cette vitesse peut être décomposée dans le repère local $(\mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3})$ de la manière suivante :

$$\mathbf{V}_j(s) = V_{tj}\mathbf{e}_{j1} + V_{nj2}\mathbf{e}_{j2} + V_{nj3}\mathbf{e}_{j3} \quad (2.52)$$

où :

V_{tj} représente la vitesse d'avance selon l'axe \mathbf{e}_{j1} ,

V_{nj2} et V_{nj3} sont les composantes de vitesse perpendiculaires selon les axes \mathbf{e}_{j2} et \mathbf{e}_{j3} ,

On définit également : $\mathbf{V}_{nj} = V_{nj2}\mathbf{e}_{j2} + V_{nj3}\mathbf{e}_{j3}$ et $\|\mathbf{V}_{nj}\| = \sqrt{V_{nj2}^2 + V_{nj3}^2}$

Des relations similaires à (2.52) peuvent être écrites pour ${}^j\dot{\mathbf{V}}_j$, ${}^j\boldsymbol{\omega}_j$ et ${}^j\dot{\boldsymbol{\omega}}_j$.

Avec cet ensemble de considérations, le modèle de contact entre le fluide et les corps de l'anguille est celui de Morison [Morison *et al.*, 1950], et peut alors être défini comme un champ de forces par unité de longueur (selon l'axe du corps) et exprimé au centre de chaque tranche.

$${}^j\mathbf{f}_{hj}(s) = \begin{bmatrix} {}^j\mathbf{f}_{hj}(s) \\ {}^j\mathbf{m}_{hj}(s) \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{f}_{drag}(s) \\ {}^j\mathbf{m}_{drag}(s) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{am}(s) \\ {}^j\mathbf{m}_{am}(s) \end{bmatrix} \quad (2.53)$$

où :

- ${}^j\mathbf{f}_{drag}(s)$ et ${}^j\mathbf{m}_{drag}(s)$ représentent respectivement les champs de forces et de moments de traînage appliqués sur la section s ;
- ${}^j\mathbf{f}_{am}(s)$ et ${}^j\mathbf{m}_{am}(s)$ représentent respectivement les champs de forces et de moments de masses ajoutées appliqués sur la section s .

Les forces et les moments de traînages sont exprimés en fonction du torseur de vitesse par :

$${}^j\mathbf{f}_{drag}(s) = C_{ld1,j} |V_{tj}(s)| V_{tj}(s)\mathbf{e}_{j1} + \sum_{i=2}^3 C_{ldi,j} \|V_{nj}(s)\| V_{nji}(s)\mathbf{e}_{ji} \quad (2.54)$$

$${}^j\mathbf{m}_{drag}(s) = C_{ad1,j} |\boldsymbol{\omega}_{tj}| \boldsymbol{\omega}_{tj}\mathbf{e}_{j1} \quad (2.55)$$

Les forces et les moments de masses ajoutées sont exprimés en fonction du torseur d'accélération par :

$${}^j\mathbf{f}_{am}(s) = \sum_{i=2}^3 C_{lmi,j} \dot{V}_{nji}(s)\mathbf{e}_{ji} \quad (2.56)$$

$${}^j\mathbf{m}_{am}(s) = C_{am1,j} \dot{\boldsymbol{\omega}}_{tj}\mathbf{e}_{j1} \quad (2.57)$$

Physiquement, \mathbf{f}_{drag} et \mathbf{m}_{drag} sont dus au frottement de l'eau (viscosité) et à la différence de pression entre les deux côtés d'un corps alors que \mathbf{f}_{am} et \mathbf{m}_{am} sont en relation avec

la quantité (ou masse) d'eau déplacée au cours du mouvement.

Pour finir, les coefficients $C_{ldi,j}$, $C_{lmi,j}$, $C_{ad1,j}$ et $C_{am1,j}$ dépendent de la masse volumique de l'eau, de la forme et de la taille du profilé (elliptique dans notre cas) et du nombre de Reynolds (approximativement 10^5). L'expression de ces termes est donnée dans le paragraphe 2.9.

Les vitesses et accélérations linéaires sont calculées par :

$${}^j\mathbf{V}_j(s) = {}^j\mathbf{V}_j + {}^j\boldsymbol{\omega}_j \times {}^j\mathbf{P}_j(s) \quad (2.58)$$

$${}^j\dot{\mathbf{V}}_j(s) = {}^j\dot{\mathbf{V}}_j + {}^j\dot{\boldsymbol{\omega}}_j \times {}^j\mathbf{P}_j(s) + {}^j\boldsymbol{\omega}_j \times {}^j\boldsymbol{\omega}_j \times {}^j\mathbf{P}_j(s) \quad (2.59)$$

où ${}^j\mathbf{P}_j(s)$ désigne la position de la tranche s par rapport à l'origine du repère du corps j . Finalement, en superposant toutes les contributions tranche par tranche de $s=0$ à $s=L_j$ (la longueur du corps j), on obtient alors la contribution totale des efforts appliqués par le corps j sur le fluide, et exprimé en \mathbf{O}_j :

$${}^j\mathbf{f}_{hj} = \begin{bmatrix} {}^j\mathbf{f}_{hj} \\ {}^j\mathbf{m}_{hj} \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{f}_{drag} \\ {}^j\mathbf{m}_{drag} \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{am} \\ {}^j\mathbf{m}_{am} \end{bmatrix} = \int_0^{L_j} {}^j\mathbf{f}_{hj}(s) ds \quad (2.60)$$

dans lequel :

$${}^j\mathbf{f}_{hj}(s) = \begin{bmatrix} {}^j\mathbf{f}_{hj}(s) \\ {}^j\mathbf{m}_{hj}(s) \end{bmatrix} = {}^s\mathbb{T}_j^T \left(\begin{bmatrix} {}^j\mathbf{f}_{drag}(s) \\ {}^j\mathbf{m}_{drag}(s) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{am}(s) \\ {}^j\mathbf{m}_{am}(s) \end{bmatrix} \right) \quad (2.61)$$

et

$${}^s\mathbb{T}_j = \begin{bmatrix} \mathbf{I}_3 & -{}^j\hat{\mathbf{P}}(s) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}$$

Le premier terme (forces de traînées et de viscosités) de l'équation (2.61) est intégré numériquement à chaque pas de calcul de l'algorithme de $s=0$ à $s=L_j$, tandis que le second terme (masses ajoutées) peut être calculé de manière explicite dans le repère local ($\mathbf{O}_j, \mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3}$) :

$${}^j\mathbf{f}_{amj} = \begin{bmatrix} {}^j\mathbf{f}_{amj} \\ {}^j\mathbf{m}_{amj} \end{bmatrix} = {}^j\mathbb{J}_j^{aj} {}^j\dot{\mathbf{V}}_j + {}^j\boldsymbol{\beta}_{amj} \quad (2.62)$$

Avec ${}^j\mathbb{J}_j^{aj}$ la matrice (6×6) d'inertie ajoutée et ${}^j\boldsymbol{\beta}_{amj}$ la matrice (6×1) des forces centrifuges et de Coriolis, tous deux introduits par la masse de fluide déplacée au cours du mouvement de l'anguille. Pour pouvoir calculer ces deux matrices, on commence par ré-écrire l'équation (2.62) dans le repère local ($\mathbf{G}_j, \mathbf{e}_{j1}, \mathbf{e}_{j2}, \mathbf{e}_{j3}$) situé au centre de masse \mathbf{G}_j du corps :

$${}^j\mathbf{f}_{amGj} = \begin{bmatrix} {}^j\mathbf{f}_{amj} \\ {}^j\mathbf{m}_{amj} \end{bmatrix} = {}^j\mathbb{J}_{Gj}^{aj} {}^j\dot{\mathbf{V}}_{Gj} + {}^j\boldsymbol{\beta}_{amGj} \quad (2.63)$$

On peut alors démontrer (sous l'hypothèse que la géométrie du corps reste constante sur tout le corps) que :

$${}^j\mathbb{J}_{Gj}^{aj} = \begin{bmatrix} \mathbf{m}_{aGj} & -\mathbf{m}\hat{\mathbf{s}}_{aGj} \\ \mathbf{m}\hat{\mathbf{s}}_{aGj} & \mathbf{I}_{aGj} \end{bmatrix} \quad (2.64)$$

et

$${}^j\boldsymbol{\beta}_{amGj}^{aj} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \frac{1}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} ({}^j\boldsymbol{\omega}_j \times {}^j\boldsymbol{\omega}_j \times {}^j\mathbf{S}_j) \end{bmatrix} \quad (2.65)$$

où ${}^j\mathbf{S}_j$ représente la position du centre de masse du corps j par rapport à l'origine du repère. En outre, en tenant compte des expressions des coefficients de masses ajoutées exprimés dans les équations (2.56) et (2.57), le détail de ${}^j\mathbb{J}_{Gj}^{aj}$ permet de retrouver le résultat exprimé dans [Susbielles et Bratuch, 1981] pour la matrice de masse ajoutée d'une ellipse fine, soit :

- une matrice (3×3) d'inertie ajoutée de translation,

$${}^j\mathbf{m}_{aGj} = \text{diag}_{i=1,2,3}(\mathbf{m}_{aij}),$$

$$\text{avec } \mathbf{m}_{a1j} = 0, \mathbf{m}_{a2j} = C_{lm2,j} \cdot L_j, \mathbf{m}_{a3j} = C_{lm3,j} \cdot L_j$$

- une matrice (3×3) de termes couplés d'inertie ajoutée de translation et de rotation,

$${}^j\mathbf{m}\hat{\mathbf{s}}_{aGj} = \mathbf{0}_{3 \times 3}$$

- une matrice (3×3) d'inertie ajoutée de rotation,

$${}^j\mathbf{I}_{aGj} = \text{diag}_{i=1,2,3}(\mathbf{I}_{aij})$$

$$\text{avec } \mathbf{I}_{a1j} = C_{am1,j} \cdot L_j, \mathbf{I}_{a2j} = C_{lm3,j} \cdot L_j^3/12, \mathbf{I}_{a3j} = C_{lm2,j} \cdot L_j^3/12$$

Ainsi, l'équation (2.64) peut être reformulée :

$$\begin{bmatrix} {}^j\mathbf{f}_{amGj} \\ {}^j\mathbf{m}_{amGj} \end{bmatrix} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^j\mathbf{I}_{aGj} \end{bmatrix} \begin{bmatrix} {}^j\dot{\mathbf{V}}_{Gj} \\ {}^j\dot{\boldsymbol{\omega}}_j \end{bmatrix} + {}^j\boldsymbol{\beta}_{amGj} \quad (2.66)$$

En exprimant ce torseur non plus au centre de masse, mais au centre du repère Σ_j , on obtient une expression détaillée de l'équation (2.61) :

$$\begin{aligned} {}^j\mathbf{f}_{hj} = & \begin{bmatrix} {}^j\mathbf{m}_{aGj} & -{}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \\ {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} & {}^j\mathbf{I}_{aGj} - {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \end{bmatrix} {}^j\dot{\mathbf{V}}_j + \\ & \begin{bmatrix} {}^j\mathbf{m}_{aGj} ({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j) \\ \left({}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} + \frac{1}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} \right) ({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{dragj} \\ {}^j\mathbf{m}_{dragj} \end{bmatrix} \end{aligned} \quad (2.67)$$

L'introduction de l'équation (2.67) dans (2.16) permet de séparer les termes correspondants au torseur ${}^j\mathbf{f}_{ej}$ en un terme dû aux forces de contact du fluide, et un terme constant de masse ajoutée sous forme d'une matrice d'inertie (6×6) . On obtient finalement :

$${}^j\mathbb{J}_j = {}^j\mathbb{J}_j + {}^j\mathbb{J}_j^{aj} \quad (2.68)$$

avec :

$${}^j\mathbb{J}_j^{aj} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} & -{}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \\ {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} & {}^j\mathbf{I}_{aGj} - {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} {}^j\hat{\mathbf{S}}_j \end{bmatrix} \quad (2.69)$$

et :

$${}^j\mathbf{f}_{ej} = \begin{bmatrix} {}^j\mathbf{m}_{aGj} ({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j) \\ \left(\frac{4}{3} {}^j\hat{\mathbf{S}}_j {}^j\mathbf{m}_{aGj} \right) ({}^j\hat{\omega}_j {}^j\hat{\omega}_j {}^j\mathbf{S}_j) \end{bmatrix} + \begin{bmatrix} {}^j\mathbf{f}_{dragj} \\ {}^j\mathbf{m}_{dragj} \end{bmatrix} \quad (2.70)$$

L'expression du modèle fluide ne dépendant plus directement des termes d'accélération, le calcul des efforts ${}^j\mathbf{f}_{ej}$ et de la matrice de masse ajoutée ${}^j\mathbb{J}_j^{aj}$ pourront être calculés pendant la première récurrence avant des algorithmes de Newton-Euler pour les modèles dynamiques inverse ou direct.

2.8 Vérification des algorithmes

Dans ce paragraphe, on valide les modèles dynamiques proposés dans ce chapitre en utilisant les théorèmes de quantité de mouvement et du moment cinétique appliqués au robot anguille et définis par :

$$\mathbb{C}_{G,S/R_g} = \begin{bmatrix} \mathbf{Q}_{S/R_g} \\ \boldsymbol{\sigma}_{G,S/R_g} \end{bmatrix}$$

Avec :

- \mathbf{Q}_{S/R_g} : le vecteur quantité de mouvement de l'anguille ;
- $\boldsymbol{\sigma}_{G,S/R_g}$: le vecteur moment cinétique de l'anguille exprimé au centre de gravité G.

Pour un corps seul C_j , ces deux vecteurs exprimés par rapport à G_j sont :

$$\mathbf{Q}_{C_j/R_g} = M_j \mathbf{V}_{G_j} \quad (2.71)$$

$$\boldsymbol{\sigma}_{G_j,C_j/R_g} = \left(\mathbf{J}_j - M_j \hat{\mathbf{S}}_j^T \hat{\mathbf{S}}_j \right) \boldsymbol{\omega}_j \quad (2.72)$$

On rappelle les deux théorèmes suivants pour la mécanique des solides :

Théorème des quantités de mouvement

Théorème: 1 *Pour tout ensemble matériel S en mouvement par rapport au repère galiléen R_g , la variation de la quantité de mouvement du système est égale à la résultante générale du torseur associé aux actions extérieures à S .*

Théorème du moment cinétique

Théorème: 2 *Pour tout ensemble matériel S en mouvement par rapport au repère galiléen R_g , la variation du moment cinétique est égale au moment résultant du torseur associé aux actions extérieures à S .*

L'application de ces deux théorèmes nous donne la relation suivante :

$$\frac{d}{dt} \mathbb{C}_{G,S/R_g} = \frac{d}{dt} \begin{bmatrix} \mathbf{Q}_{S/R_g} \\ \boldsymbol{\sigma}_{G,S/R_g} \end{bmatrix} = \sum \mathbb{F}_{G,\bar{S} \rightarrow S} \quad (2.73)$$

avec : $\mathbb{F}_{G,\bar{S} \rightarrow S} = \begin{bmatrix} \mathbf{F}_{\bar{S} \rightarrow S} & \mathbf{M}_{G,\bar{S} \rightarrow S} \end{bmatrix}^T$

On suppose à présent que l'anguille évolue dans le vide, ce qui signifie qu'il n'existe aucune force et aucun moment extérieur. Par conséquent :

$$\frac{d}{dt} \mathbb{C}_{G,S/Rg} = \mathbf{0} \quad (2.74)$$

soit pour tout instant t :

$$\begin{bmatrix} \mathbf{Q}_{S/Rg} \\ \boldsymbol{\sigma}_{G,S/Rg} \end{bmatrix} = C^{ste} \quad (2.75)$$

A $t=0$ de la simulation, on suppose que l'anguille n'a aucune loi de mouvements interne et qu'elle est immobile dans l'espace. Avec cette hypothèse de départ, on peut en déduire que pour $t=0$ puis pour tout instant t :

$$\begin{bmatrix} \mathbf{Q}_{S/Rg} \\ \boldsymbol{\sigma}_{G,S/Rg} \end{bmatrix} = \mathbf{0} \quad (2.76)$$

De ce résultat, on peut en déduire qu'à tout instant, le centre de gravité de l'anguille ne bougera pas et qu'il n'y aura aucun mouvement de rotation autour de celui-ci. De plus, lorsque le vecteur quantité de mouvement $\mathbf{Q}_{S/Rg}$ est nul, on peut alors calculer le torseur cinétique en n'importe quel point de l'espace (le vecteur moment cinétique demeure dans ce cas constant quel que soit le point d'application).

L'anguille étant constituée d'un ensemble de corps rigides en mouvement les uns par rapport aux autres, on calcule le torseur cinétique pour chacun d'eux, puis on en fait la somme en les exprimant au centre d'un repère fixe Σ_w . Au final, on obtient le torseur cinétique de toute l'anguille.

2.8.1 Calcul du torseur cinétique de l'anguille

Soit Σ_w un repère galiléen fixe dans lequel l'anguille évolue. Le torseur cinétique de l'ensemble des corps de l'anguille est alors défini de la manière suivante :

$$\begin{aligned} {}^w\mathbb{C}_{S/\Sigma_w} &= \begin{bmatrix} {}^w\mathbf{Q}_{S/\Sigma_w} \\ {}^w\boldsymbol{\sigma}_{O,S/\Sigma_w} \end{bmatrix} \\ &= \sum_{j=0}^n \left[{}^w\mathbf{M}\mathbf{S}_j \times ({}^w\mathbf{R}_j {}^j\mathbf{V}_{Gj}) + {}^w\mathbf{R}_j \left({}^j\mathbf{J}_j - M_j \hat{\mathbf{S}}_j^T \hat{\mathbf{S}}_j \right) {}^j\boldsymbol{\omega}_j \right] \end{aligned} \quad (2.77)$$

avec :

$${}^j\mathbf{V}_{Gj} = {}^j\mathbf{V}_j - {}^j\hat{\mathbf{S}}_j {}^j\boldsymbol{\omega}_j \quad (2.78)$$

$${}^w\mathbf{MS}_j = M_j ({}^w\mathbf{P}_j + {}^w\mathbf{R}_j {}^j\mathbf{S}_j) \quad (2.79)$$

où :

- ${}^w\mathbf{P}_j$ représente la position du centre de l'articulation O_j du corps j dans le repère Σ_w ,
- ${}^j\mathbf{S}_j$ représente la position du centre de gravité du corps j par rapport au centre du repère Σ_j .

En développant l'équation (2.77), on obtient :

$${}^w\mathbb{C}_{S/\Sigma_w} = \sum_{j=0}^n \left[{}^w\mathbf{R}_j M_j - {}^w\mathbf{R}_j M_j {}^j\hat{\mathbf{S}}_j - M_j {}^w\mathbf{P}_j \times ({}^w\mathbf{R}_j {}^j\hat{\mathbf{S}}_j) + {}^w\mathbf{R}_j {}^j\mathbf{J}_j \right] {}^j\mathbb{V}_j \quad (2.80)$$

ou encore :

$${}^w\mathbb{C}_{S/\Sigma_w} = \sum_{j=0}^n \begin{bmatrix} {}^w\mathbf{R}_j & \mathbf{0}_{3 \times 3} \\ {}^w\hat{\mathbf{P}}_j & {}^w\mathbf{R}_j \end{bmatrix} \begin{bmatrix} M_j \mathbf{I}_3 & -M_j {}^j\hat{\mathbf{S}}_j \\ M_j {}^j\hat{\mathbf{S}}_j & {}^j\mathbf{J}_j \end{bmatrix} {}^j\mathbb{V}_j \quad (2.81)$$

On retrouve ici l'expression de deux matrices connues tel que :

$${}^w\mathbb{C}_{S/\Sigma_w} = \sum_{j=0}^n {}^j\mathbb{T}_w^T {}^j\mathbb{J}_j {}^j\mathbb{V}_j \quad (2.82)$$

Le calcul du torseur cinétique peut donc se faire en utilisant l'algorithme suivant :

1) On initialise la matrice de transformation entre torseur ${}^0\mathbb{T}_w^T$:

$${}^0\mathbb{T}_w^T = \begin{bmatrix} {}^w\mathbf{R}_0 & \mathbf{0}_{3 \times 3} \\ {}^w\hat{\mathbf{P}}_0 & {}^w\mathbf{R}_0 \end{bmatrix}$$

Avec :

- ${}^w\mathbf{R}_0$ la matrice de rotation de la tête par rapport au référentiel galiléen Σ_w .
- ${}^w\mathbf{P}_0$ le vecteur de position de la tête par rapport au centre du repère galiléen Σ_w .

Puis on initialise le calcul du torseur cinétique :

$${}^w\mathbb{C}_0 = {}^0\mathbb{T}_w^T {}^0\mathbb{J}_0 {}^0\mathbb{V}_0$$

2) On calcule ensuite la matrice de transformation et le torseur cinétique pour $j=1, \dots, n$:

$${}^j\mathbb{T}_w^T = {}^{j-1}\mathbb{T}_w^T {}^j\mathbb{T}_{j-1}^T \quad (2.83)$$

$${}^w\mathbb{C}_j = {}^w\mathbb{C}_{j-1} + {}^j\mathbb{T}_w^T {}^j\mathbb{J}_j {}^j\mathbb{V}_j \quad (2.84)$$

2.8.2 Générateur de mouvements

Pour pouvoir simuler le comportement de l'anguille, on utilise un générateur de mouvements classiquement utilisé dans la littérature [Gillis, 1998, Tytell et Lauder, 2004, Boyer *et al.*, 2006b]. Ce générateur a pour but de calculer les consignes articulaires que l'on va appliquer au robot afin que celui puisse nager. Il est basé sur une fonction sinusoïdale qui permet d'engendrer une forme ondulatoire de la nage de l'anguille. La liaison entre chaque segment de l'anguille étudiée étant de type rotule, on décompose les différents mouvements sous forme de trois fonctions. On distingue une première fonction qui permet de générer l'onde sinusoïdale créant la propulsion de l'anguille. Les deux autres fonctions servent respectivement à contrôler l'altitude et le roulis de l'anguille.

La propulsion de l'anguille s'obtient à partir de l'onde sinusoïdale qui parcourt son corps de la tête vers la queue (et vice versa si l'anguille veut reculer). Cette onde sinusoïdale est paramétrée par sa période d'oscillation et sa longueur d'onde. Pour l'amplitude de l'onde générée, plus on va vers la queue, plus l'anguille augmente l'amplitude afin d'améliorer l'efficacité de la nage.

Générateur de propulsion :

La propulsion de l'anguille est produite en utilisant une fonction sinusoïdale continue de la forme suivante qu'on applique sur les articulations $3j-2$ ($j=1, \dots, 12$) du robot :

$$Q(s, t) = f_1(t) \cdot A \cdot e^{\alpha s} \cdot \sin \left[2\pi \left(\frac{s}{\lambda} - \frac{t}{T} \right) \right] \quad (2.85)$$

Pour laquelle :

- A : amplitude du mouvement ;
- α : réglage de l'exponentielle : augmentation de l'amplitude de la tête vers la queue ;
- T : période de l'oscillation ;
- λ : longueur d'onde de l'oscillation ;
- s : abscisse curviligne le long de la fibre neutre de l'anguille ;
- $f_1(t)$: fonction de lissage du démarrage et de l'arrêt.

Pour appliquer cette loi de mouvement continue, on est obligé de discrétiser l'équation (2.85) pour obtenir les valeurs à appliquer sur la première articulation $3j-2$ ($j=1, \dots, 12$) de chaque rotule du robot [Yu *et al.*, 2005].

$$q_{3j-2}(t) = Q(X_{j+1}, t) - Q(X_j, t) + f_2(t)O_{ff1} \quad (2.86)$$

où

- $X_j = \sum_{k=1}^j r_k$ est la distance de chaque liaison rotule par rapport à l'origine de la tête ;
- $f_2(t)$: fonction pour amorcer les virages ;
- O_{ff1} : "offset" pour régler la courbure de l'anguille afin de faire des virages.

Réglage de l'altitude :

Pour que l'anguille puisse changer d'altitude (ou de profondeur) pendant sa nage, on impose sur la deuxième articulation $3j-1$ ($j=1, \dots, 12$) de chaque rotule la fonction suivante :

$$q_{3j-1} = f_3(t).O_{ffs2} \quad (2.87)$$

On se contente dans l'état actuel des simulations d'imposer un "offset" O_{ffs2} constant sur les articulations pour faire des trajectoires de type looping ou plongée. Cet "offset" peut être ajusté par une fonction de démarrage et d'arrêt $f_3(t)$ semblable à $f_1(t)$.

Réglage du roulis :

Pour que l'anguille puisse effectuer une rotation propre autour de l'axe de la tête, on impose sur la troisième articulation $3j$ ($j=1, \dots, 12$) de chaque rotule la fonction suivante :

$$q_{3j} = f_4(t).O_{ffs3} \quad (2.88)$$

Ce terme permet de régler la torsion de l'anguille afin d'agir sur le roulis de la tête (Ce terme est plus utile pour un fonctionnement de type serpent dans une canalisation par exemple). On se contente donc, tout comme dans le cas précédent, d'imposer un simple "offset" O_{ffs3} constant qui peut être lissé par la fonction $f_4(t)$.

Expression des fonctions $f_1(t)$ à $f_4(t)$:

Les fonctions $f_1(t)$ à $f_4(t)$ sont des fonctions polynomiales du 5ème degré :

$$\begin{aligned} f_i(t) &= f_i(t) = a_0.t^5 + a_1.t^4 + a_2.t^3 + a_3.t^2 + a_4.t + a_5 & t \leq t_{fi} \\ f_i(t) &= 1 & t > t_{fi} \end{aligned} \quad (2.89)$$

Les coefficients a_i de ces polynômes satisfont les conditions aux limites suivantes :

$$f_i(0) = 0, f_i(t_{fi}) = 1, \dot{f}_i(0) = \ddot{f}_i(0) = 0, \dot{f}_i(t_{fi}) = \ddot{f}_i(t_{fi}) = 0$$

où t_{fi} représente le temps de fin de $f_i(t)$.

2.8.3 Validation des calculs

Pour valider les algorithmes que nous avons présentés, on réalise une simulation du robot en utilisant le modèle dynamique inverse en considérant qu'il n'y a pas d'efforts extérieurs appliqués sur l'anguille (simulation dans le vide), puis on calcule pour chaque pas de temps de la simulation le torseur cinétique. A partir de ces résultats, nous avons démontré que pour une simulation dans le vide le torseur cinétique devait être nul. Pour le vérifier, on impose que pour tous les corps de l'anguille ($j=0,\dots,n$) ${}^j\mathbf{f}_{ej} = 0$ (pas d'efforts extérieurs) et ${}^j\mathbb{J}_j^a = 0$ (pas de masses ajoutées), puis on effectue une simulation en utilisant les paramètres suivants pour le générateur de mouvements :

$$A=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; t_f=3 \text{ s} ; \lambda=1,5 \text{ m} ; T=3 \text{ s} ; O_{ff_1}=0 ; O_{ff_2}=0 ; O_{ff_3}=0$$

La figure (2.6) montre une configuration de l'anguille pendant la simulation. Ensuite, les figures (2.7) et (2.8) montrent respectivement les 3 composantes de la quantité de mouvement et du moment cinétique obtenus pendant cette simulation :

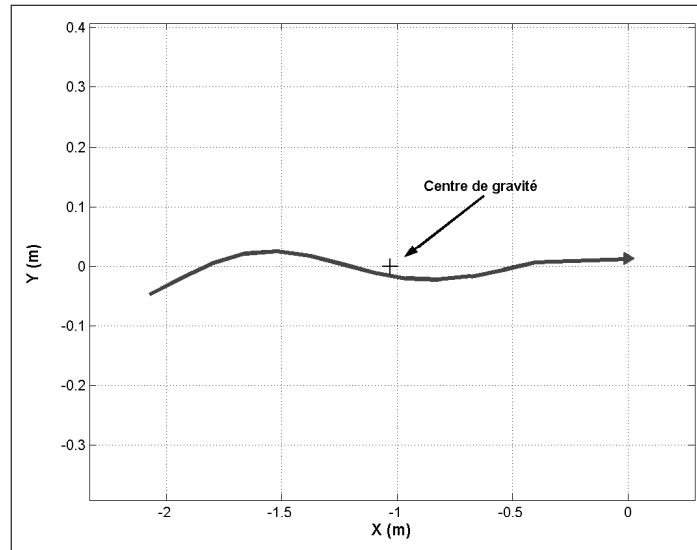


Figure 2.6: – Une configuration de l'anguille pendant la simulation dans le vide

L'anguille sur la figure (2.6) est représentée en mode filaire (on trace uniquement le centre de la colonne vertébrale). Le centre de gravité est représenté par la croix et la tête de l'anguille est représentée par la flèche. Pour rappel, au début de la simulation, le repère de la tête de l'anguille Σ_0 et le repère fixe Σ_w sont confondus.

Pour vérifier que le centre de gravité ne bouge pas, on trace les 3 composantes du vecteur quantité de mouvement ainsi que les 3 composantes du moment cinétique de l'anguille au cours du mouvement et on vérifie qu'elles sont bien nulles durant toute la durée de la simulation :

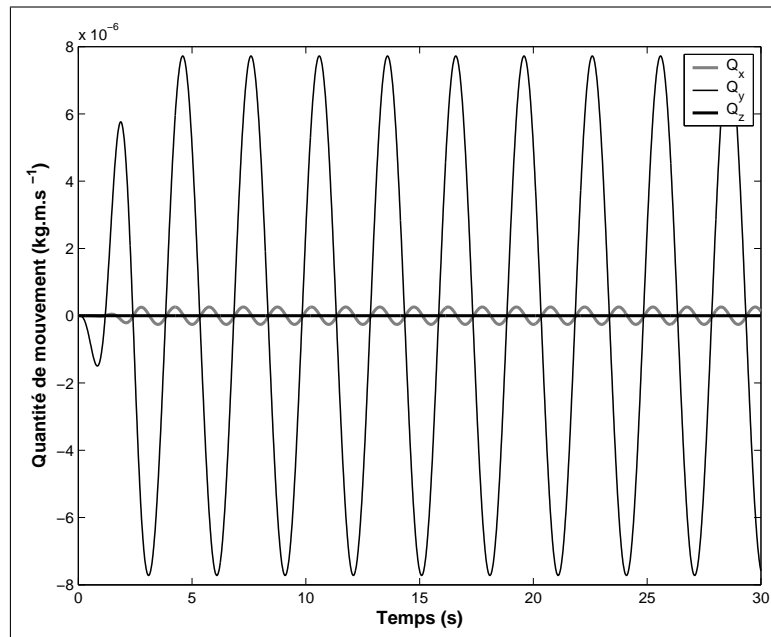


Figure 2.7: – Vecteur Quantité de mouvement de l'anguille

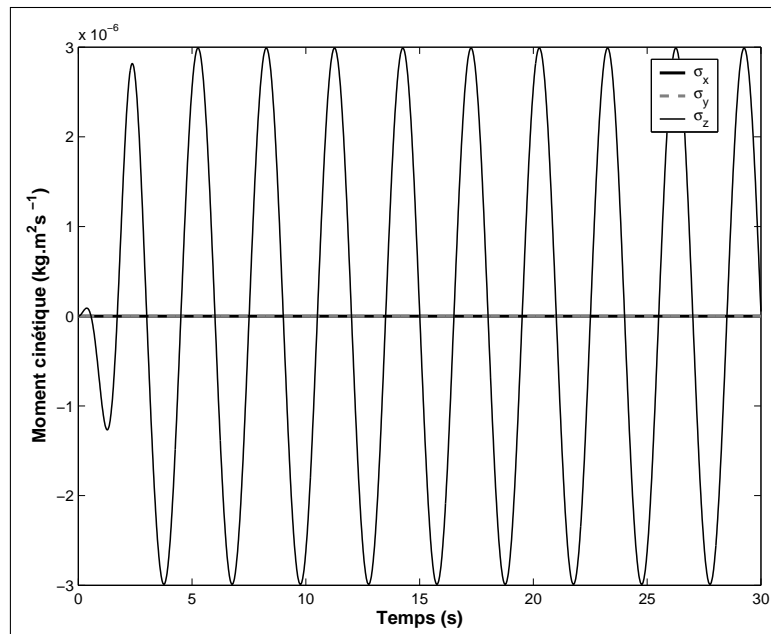


Figure 2.8: – Vecteur Moment cinétique de l'anguille

Sur ces deux figures, on voit que pour toute la durée de la simulation, les 6 composantes du torseur cinétique sont bien nulles (de l'ordre de 10^{-6}). L'algorithme de Newton-Euler utilisé respecte donc bien les propriétés mécaniques d'un système multi-corps qui n'est soumis à aucune force extérieure.

2.8.4 Validation croisée des résultats

Cette validation, dite "croisée", permet de vérifier la cohérence entre les modèles dynamiques inverse et direct. Pour effectuer cette validation, une simulation est effectuée en utilisant le modèle dynamique inverse pour laquelle on spécifie $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, $\ddot{\mathbf{q}}(t)$ comme entrées et ${}^0\mathbf{V}_0 = 0$ comme condition initiale. Ensuite, on utilise le modèle dynamique direct en utilisant comme entrées $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ et $\mathbf{\Gamma}(t)$ obtenu par le modèle dynamique inverse et la condition initiale ${}^0\mathbf{V}_0 = 0$. Une fois le calcul des deux modèles dynamiques effectué, on doit vérifier que :

- l'accélération de la tête est la même que ce soit en utilisant le modèle dynamique inverse ou le modèle dynamique direct,
- les accélérations articulaires calculées par le modèle dynamique direct sont les mêmes que les accélérations du générateur de mouvement pour le modèle dynamique inverse.

Le schéma Simulink permettant de réaliser cette validation est donné sur la figure (2.9) :

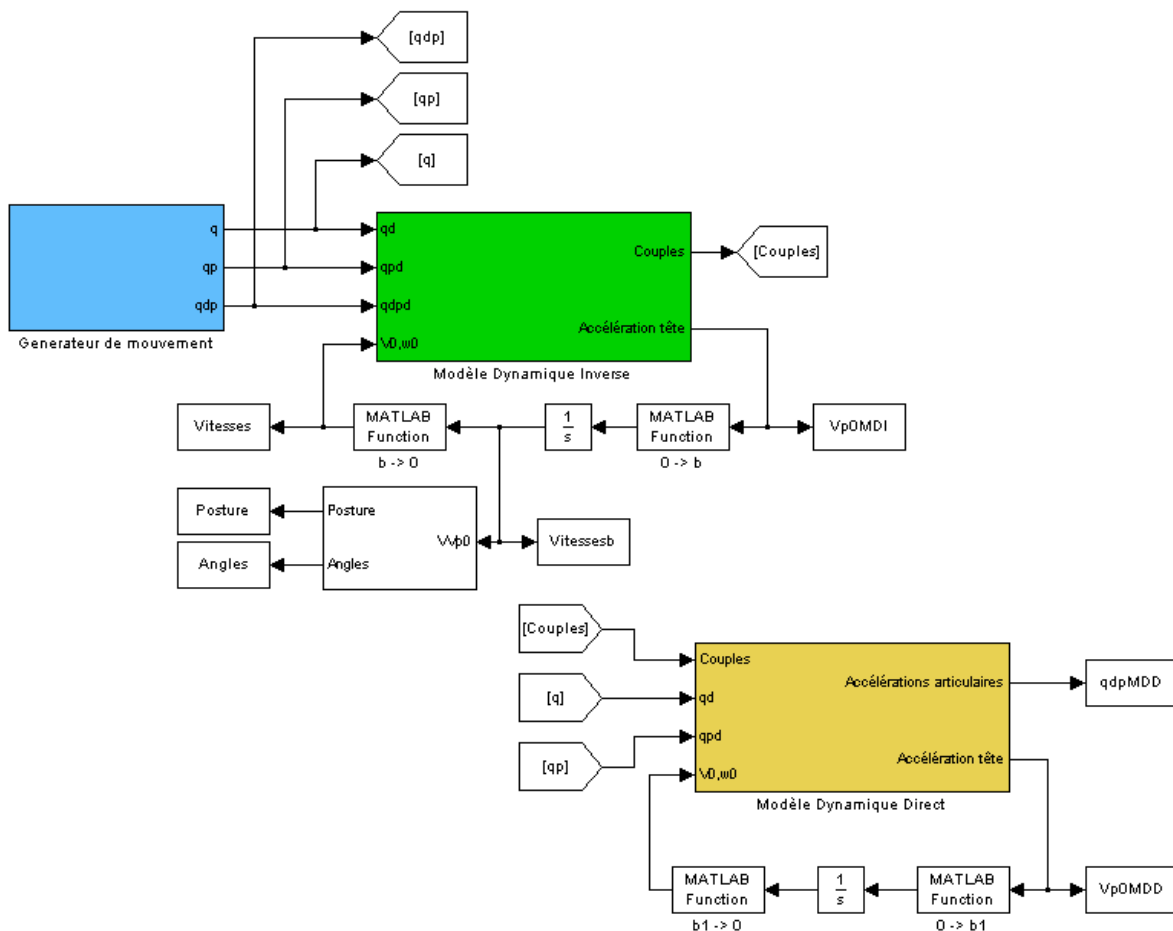


Figure 2.9: – Simulateur de la validation croisée

Les figures (2.10 -a) et (2.10 -b) montrent respectivement la comparaison de l'accélération de la tête calculée avec les deux algorithmes et la comparaison entre les accélérations articulaires calculées par le modèle dynamique inverse et les accélérations imposées au modèle dynamique inverse.

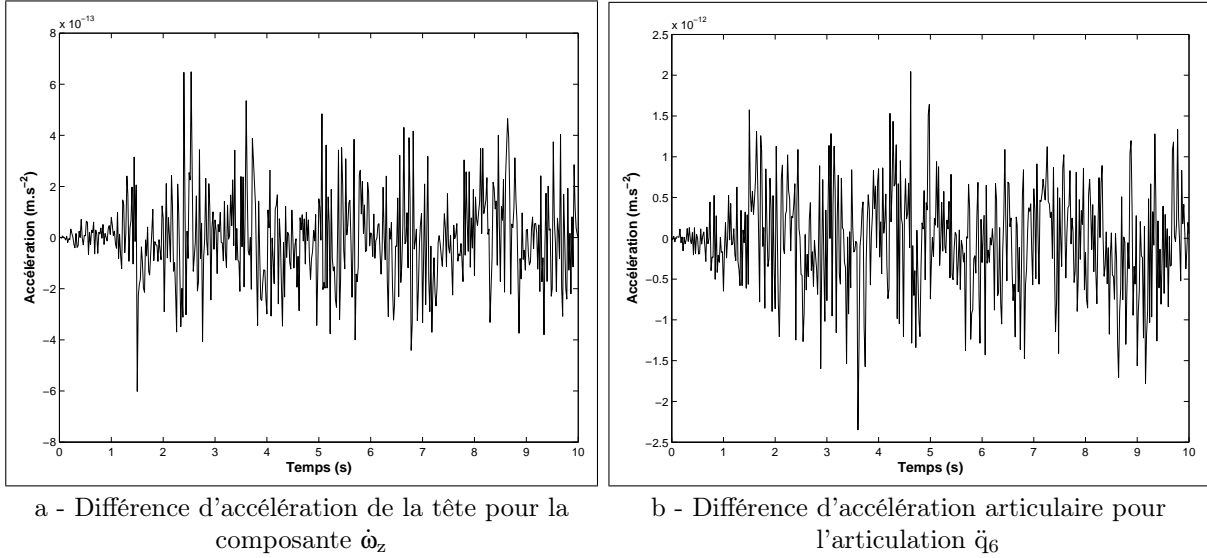


Figure 2.10: – Différence des accélérations entre le MDI et le MDD

Sur ces deux figures, on remarque que les différences obtenues sont assimilables à du bruit numérique (de l'ordre de 10^{-12} , 10^{-13}). Pour les deux figures, on a uniquement tracé la composante présentant la plus grande différence entre les résultats des deux simulations. Ces résultats de simulation nous montrent donc que la cohérence entre les deux algorithmes est bien vérifiée, et valide une nouvelle fois les algorithmes développés.

2.9 Exemples de simulations

Dans ce paragraphe, on présente quelques résultats de simulations obtenus pour le robot anguille [Khalil *et al.*, 2005] en utilisant le logiciel Matlab-Simulink . Pour pouvoir utiliser le modèle fluide, on a vu dans le paragraphe 2.7 que les coefficients C_{ldj} , C_{lmj} , C_{am1} et C_{ad1} dépendent de la géométrie de l'anguille :

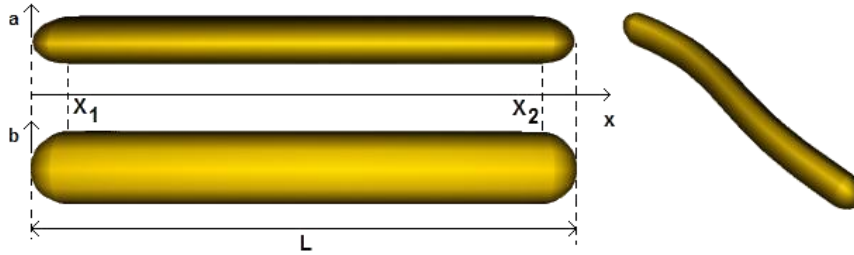


Figure 2.11: – Géométrie de l'anguille

Pour pouvoir exprimer ces paramètres, on note $2a$ et $2b$ les petit et grand axes de l'ellipse définissant la section de l'anguille (voir tableau (2.2) pour les valeurs de a et de b). D'après la littérature [Susbielles et Bratuch, 1981], les coefficients du modèle fluide-structure (2.54)-(2.57) sont les suivants :

- $C_{ld1} = (1/2)\rho C_1\pi(a + b)/2$
- $C_{ld2} = (1/2)\rho C_2 2b$
- $C_{ld3} = (1/2)\rho C_3 2a$
- $C_{ad1} = (1/2)\rho C_4 (b^2 - a^2)^2$
- $C_{lm2} = \rho\pi b^2 C_5$
- $C_{lm3} = \rho\pi a^2 C_6$
- $C_{am1} = \rho\pi C_7 (b^2 - a^2)^2 / 8$

Avec ρ la masse volumique du robot prise égale à celle de l'eau.

Et pour valeurs numériques $C_1 = 0,01$, $C_2 = C_3 = C_4 = 1$ et $C_5 = C_6 = C_7 = 1$, ce qui correspond aux valeurs pour un objet de type cylindrique plongé dans un écoulement avec un nombre de Reynolds approximatif de $Re \approx 10^5$, soit une anguille nageant à une vitesse d'environ 1 ms^{-1} .

2.9.1 Premier exemple : la nage plane

En se référant à la littérature bio-mécanique sur la locomotion anguilliforme [Gillis, 1998, Tytell et Lauder, 2004, Boyer *et al.*, 2006b], on commence par traiter le cas de la propulsion en ligne droite dans un plan.

Pour la simulation, on utilise la configuration initiale suivante :

- le repère de la tête Σ_0 est confondu avec le repère de référence Σ_w ;
- la vitesse initiale de l'anguille est nulle ;
- l'anguille est droite ($\forall j, q_j = 0$) et alignée suivant l'axe \mathbf{x}_w du repère Σ_w .

Pour cette simulation de nage en ligne droite, les 3 fonctions du générateur de mouvements sont paramétrées en utilisant les valeurs numériques suivantes :

$$A=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; t_f=3 \text{ s} ; \lambda=1,5 \text{ m} ; T=3 \text{ s} ; O_{ff_1}=0 ; O_{ff_2}=0 ; O_{ff_3}=0$$

Avec ces paramètres, on remarque que les fonctions de réglage de l'altitude et de roulis donnent à tout instant t des consignes nulles pour les articulations $3j-1$ et $3j$, $j=1,\dots,12$. La fonction du générateur de propulsion permet quant à elle de générer une loi de mouvements pour les articulations $3j-2$ et qui donne pour $j=4, 7$ et 10 les formes d'onde présentées sur la figure (2.12) :

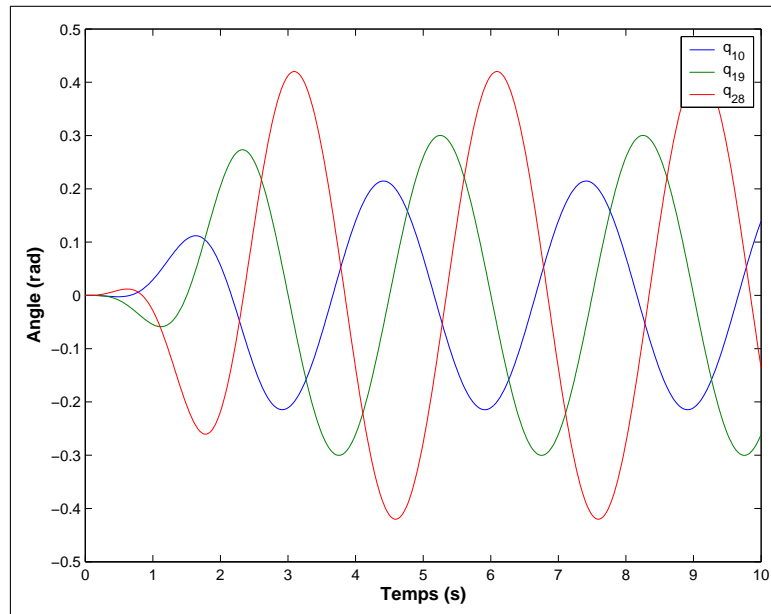


Figure 2.12: – Loi horaire des articulations $3j-2$ servant à la propulsion pour $j=4, 7$ et 10

Cette figure nous montre que $f_1(t)$ permet d'avoir une augmentation progressive de l'ondulation pendant les premières secondes de la simulation. Elle nous montre également que l'amplitude des oscillations augmente quand j augmente (on va de la tête vers la queue).

Le mode de fonctionnement souhaité au départ est donc respecté avec ce générateur de mouvements.

La figure (2.13) montre la trajectoire de la tête de l'anguille dans le plan $\mathbf{x}_w\text{-}\mathbf{y}_w$ du repère fixe pour une simulation de 30 s.

La figure (2.14) montre la vitesse absolue $\sqrt{V_x^2 + V_y^2}$ du robot dans le repère Σ_w .

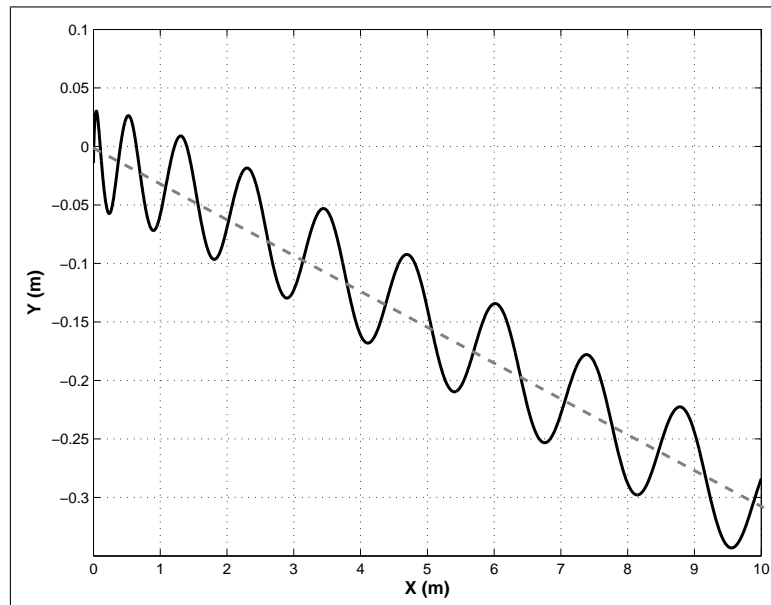


Figure 2.13: – Trajectoire de la tête dans le plan $\mathbf{x}_w\text{-}\mathbf{y}_w$ du repère fixe

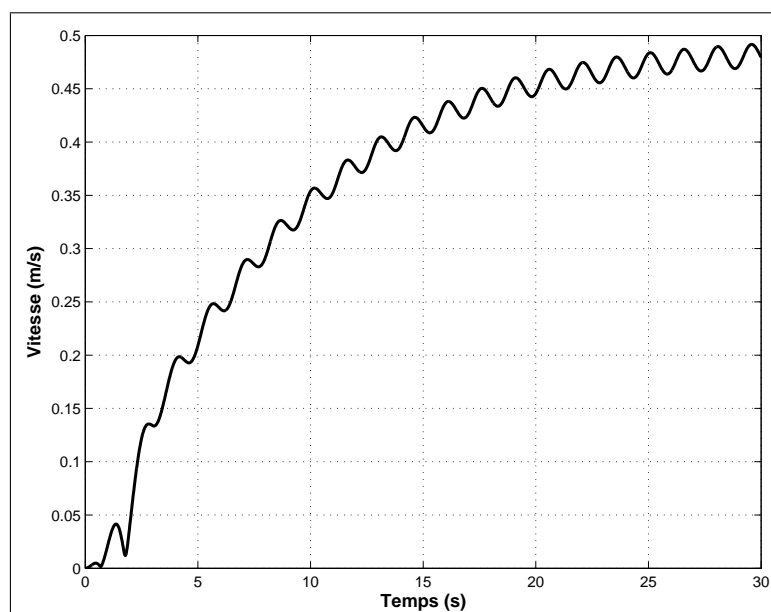


Figure 2.14: – Vitesse absolue de la tête par rapport au temps

On peut également tracer le couple obtenu par le modèle dynamique inverse pour plusieurs articulations :

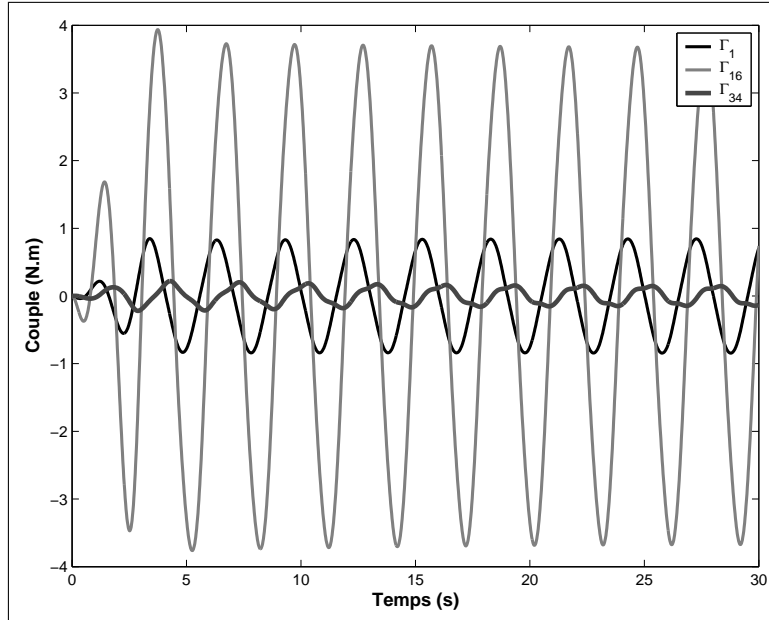


Figure 2.15: – Couples moteurs des articulations 1,16 and 34

Ces résultats montrent que l'anguille effectue bien une trajectoire en ligne droite avec une légère oscillation transverse. On observe sur la figure (2.13) que la tête de l'anguille n'oscille pas autour de l'axe des x mais autour d'un axe dévié (axe en pointillés gris). Cette déviation est due à la fonction $f_1(t)$ qui induit une dissymétrie du mouvement pendant les 3 premières secondes qui, par conséquent, induit des efforts de poussées dissymétriques. Avec les valeurs de coefficients proposés, on note que l'anguille a besoin d'environ 30 secondes pour atteindre sa vitesse de croisière soit environ 0,48 m/s. Finalement, on remarque que la courbe de vitesse de l'anguille (Figure 2.14) peut être approximée par un système du premier ordre, ce qui peut s'avérer pratique pour la synthèse de lois de commandes.

Par le choix de différents paramètres de simulation, il est possible de faire en sorte que l'anguille nage avec une vitesse plus rapide et avec un temps de réponse plus court. Cependant, dans le cadre de ces simulations, il faut prendre en compte la limitation en couple des actionneurs (actionneurs choisis dans le cadre du projet). Ainsi, les valeurs limites à ne pas dépasser sont pour $j=1,\dots,12$:

$\Gamma_{\max} = \pm 4 \text{ N.m}$ pour les articulations $3j-2$ et $3j-1$,

$\Gamma_{\max} = \pm 0.7 \text{ N.m}$ pour les articulations $3j$.

2.9.2 Deuxième exemple : nage 3D en looping

Le but de cet exemple est de faire en sorte que l'anguille fasse une trajectoire de type cercle dans le plan $\mathbf{x}_w\text{-}\mathbf{z}_w$ du repère fixe. Ce type de trajectoire peut être obtenue en utilisant de manière simultanée la fonction de générateur de mouvements et la fonction de réglage d'altitude. Pour rappel, cette fonction d'altitude joue sur la loi d'évolution des secondes articulations des liaisons rotules (articulations $3j-1$ pour $j=1,\dots,12$) et peut être paramétrée sous la forme :

$$q_{3j-1} = f_3(t) \cdot O_{ff_2}$$

où $f_3(t)$ représente une fonction polynomiale du 5ème degré définie précédemment. Pour cette simulation de nage en looping, les 3 fonctions du générateur de mouvements sont paramétrées en utilisant les valeurs numériques suivantes :

$$A=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; t_f=3 \text{ s} ; \lambda=1,5 \text{ m} ; T=3 \text{ s} ; O_{ff_1}=0 ; O_{ff_2}=\frac{\pi}{40} ; O_{ff_3}=0$$

On remarque que les paramètres pour la fonction de générateur de mouvements et pour le réglage du Roulis ont été conservés.

La fonction de réglage d'altitude qui est à présent utilisée, permet d'imposer à toutes les secondes articulations des liaisons rotules (articulations $3j-1$ pour $j=1,\dots,12$) la loi de mouvements représentée sur la figure (2.16) :

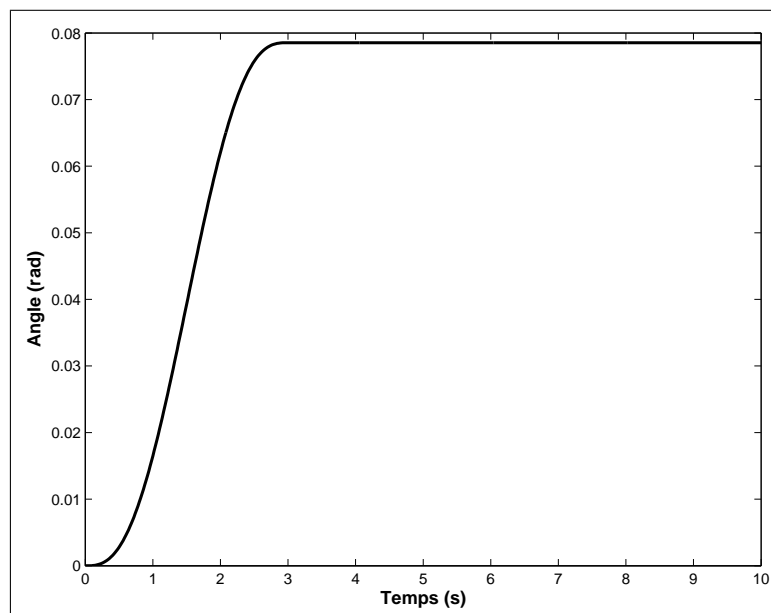


Figure 2.16: – Loi horaire des articulations $3j-1$ permettant le réglage en altitude

Les figures (2.17) à (2.19) montrent respectivement les couples articulaires des articulations $3j-2$, $3j-1$ et $3j$ (qui constituent une rotule) pour $j=1,9$ et 12 . Les figures (2.20 -a) et (2.20 -b) représentent la trajectoire de la tête dans le plan $\mathbf{x}_w\text{-}\mathbf{z}_w$ du repère fixe, et 10 configurations de l'anguille durant la simulation. Les résultats sont obtenus pour une simulation d'une durée de 60 secondes.

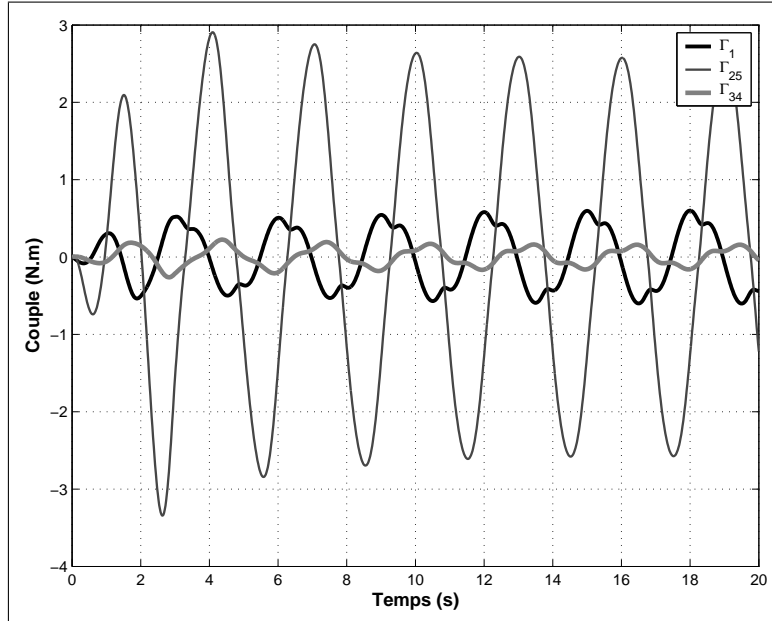


Figure 2.17: – Couples moteurs des articulations 1, 25 et 34

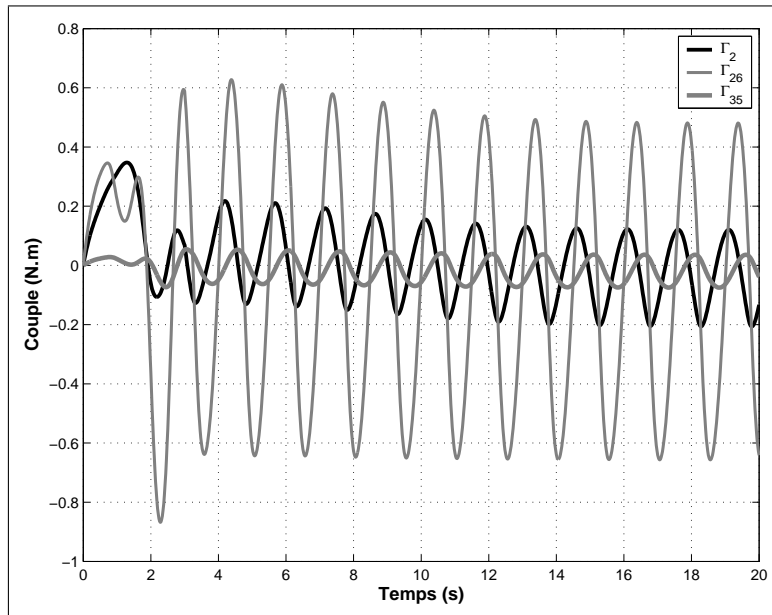


Figure 2.18: – Couples moteurs des articulations 2, 26 et 35

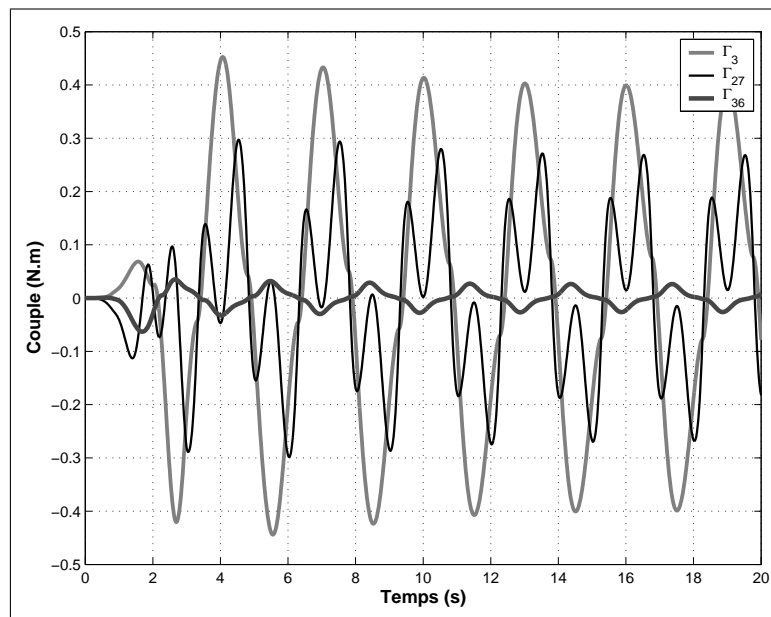


Figure 2.19: – Couples moteurs des articulations 3, 27 et 36

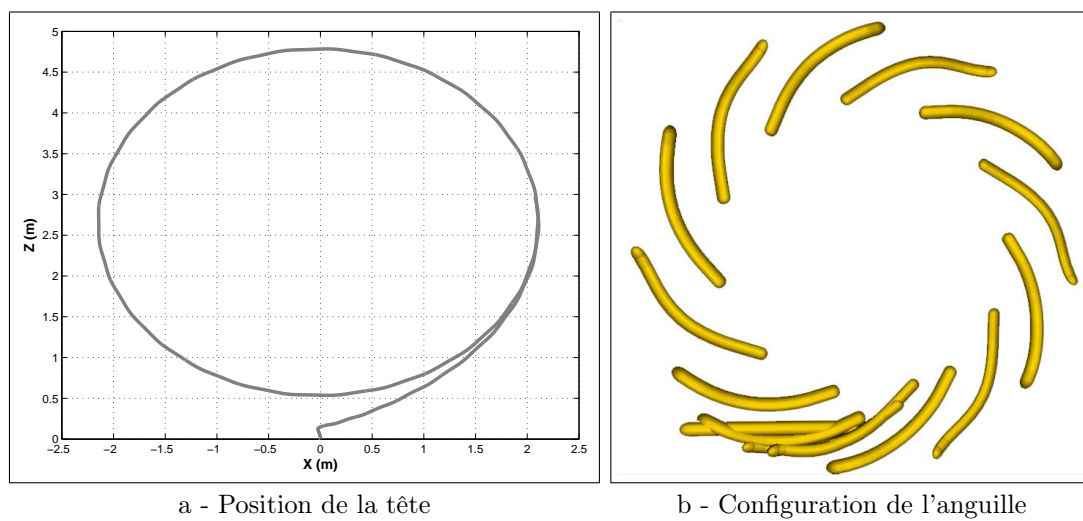


Figure 2.20: – Trajectoire de la tête de l'anguille dans le plan $\mathbf{x}_w\text{-}\mathbf{z}_w$ du repère fixe et configuration de l'anguille toutes les 6 secondes

On voit sur la figure (2.20) que la trajectoire réalisée par la tête de l'anguille est bien un cercle dans le plan $\mathbf{x}_w\text{-}\mathbf{z}_w$. Ce deuxième exemple montre donc que l'utilisation de la fonction d'altitude permet bien de faire varier l'altitude de l'anguille pendant la nage et valide donc les possibilités des simulations dans un espace 3D.

2.9.3 Analyse des résultats

On a pu voir que les résultats de simulation étaient probants aussi bien pour la nage plane que pour la nage 3D. Cependant, en dehors de ce côté qualitatif, plusieurs points peuvent être abordés afin d'apporter un regard critique sur les résultats obtenus.

Tout d'abord, en ce qui concerne le choix des paramètres de la loi d'ondulation, hormis celui de la fonction d'ondulation, on peut se poser la question du choix des valeurs numériques et de leurs pertinences. Pour répondre à cette question, nous allons regarder l'influence des paramètres A et α sur les performances de l'anguille. Pour les autres paramètres, on prendra les valeurs suivantes :

$$t_f=3s, \lambda=1,5m, T=3s, O_{ff_1}=0, O_{ff_2}=0, O_{ff_3}=0$$

On effectue une simulation de nage en ligne droite (en utilisant le générateur de mouvement de l'équation (2.85)) puis on trace en fonction de α la vitesse que peut atteindre l'anguille pour différentes valeur de A (figure 2.21) :

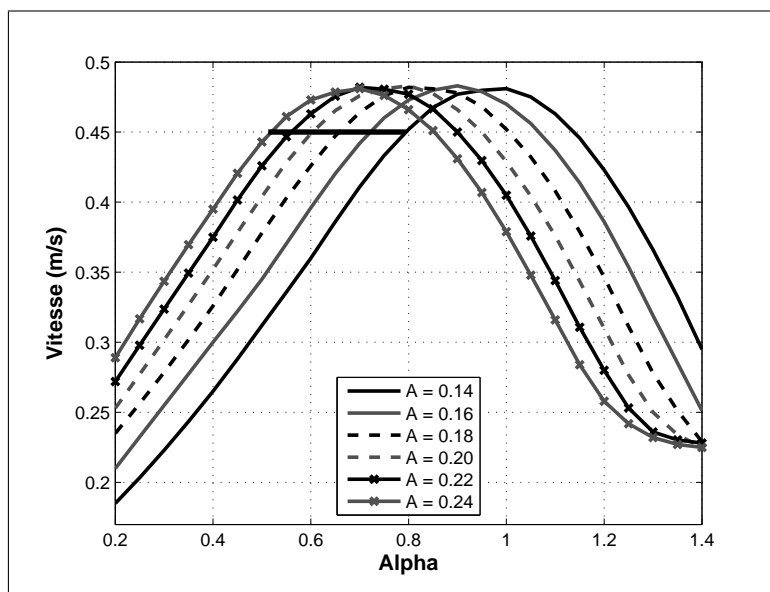


Figure 2.21: – Vitesse de la tête en fonction de α et pour différentes valeurs de A

L'analyse de la figure (2.21) nous montre que pour les différentes valeurs de A choisies, la vitesse maximale qu'il est possible d'atteindre est toujours la même (ici 0,48 m/s avec les coefficients λ et T utilisés). De même, pour une vitesse donnée que l'on souhaite atteindre, on remarque qu'il existe une multitude de couples (A, α) permettant d'y arriver (pour une valeur de A il existe déjà deux valeurs de α).

Pour savoir quelle valeur de couple choisir, un critère intéressant peut être de regarder la valeur des couples moteurs correspondants afin de minimiser l'énergie. Pour chaque

couple de valeur (A, α) , on calcule grâce au modèle dynamique inverse les couples articulaires Γ_i , puis on détermine parmi toutes les articulations quelle est la valeur maximale $\Gamma_{\max} = \max(\Gamma_i)$. On trace ensuite sur la figure (2.22) cette valeur maximale des couples en fonction de α et pour différentes valeurs de A .

Pour illustrer le problème du choix du couple (A, α) , nous avons choisi que la vitesse à atteindre pour l'anguille devait être 0,45 m/s. A partir de cette vitesse et pour chaque valeur de A , la figure (2.21) nous permet de déterminer la valeur de α correspondante (ligne noire en trait fort sur la figure 2.21). Nous avons ensuite reporté le correspondant de cette ligne sur la courbe des couples (figure 2.22) à partir des α calculés :

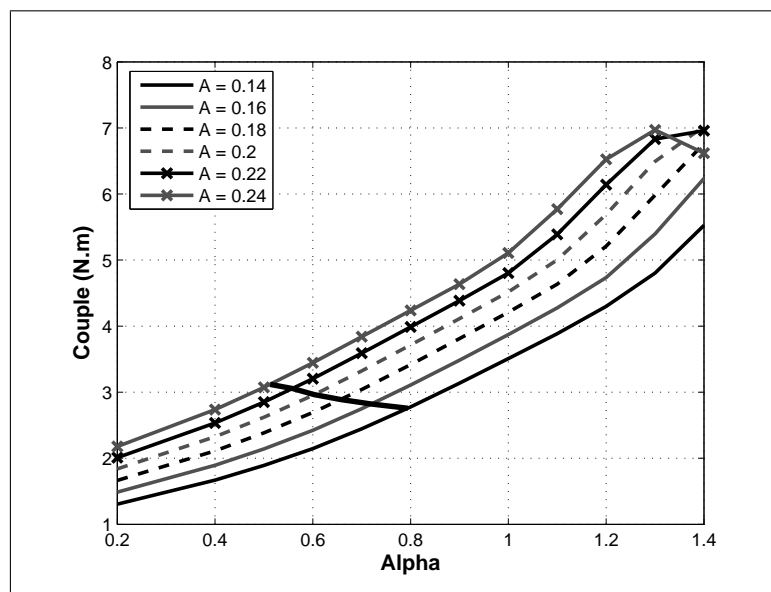


Figure 2.22: – Couple moteur maximum en fonction de α et pour différentes valeurs de A

On remarque sur la figure (2.22) que les courbes de couples sont monotones et que les valeurs augmentent en même temps que α . Le tracé de notre droite (ligne noire en trait fort), correspondant à une vitesse de l'anguille de 0,45 m/s, nous montre qu'il est plus judicieux de prendre la valeur $A=0.14$ que $A=0.24$ puisque pour une même vitesse le couple maximum nécessaire est plus faible.

Cette petite étude nous montre que connaître l'influence des différents paramètres de nage permet de choisir un jeu de valeurs optimal par rapport aux objectifs fixés. Nous n'avons fait varier ici que A et α et il pourrait être intéressant de généraliser cette approche pour l'ensemble des paramètres de nage.

On peut à présent se poser la question de la pertinence des résultats de la nage 3D. Ce type de nage est en réalité assez complexe puisqu'elle combine un mouvement d'on-

dulation dans le plan horizontal de l'anguille avec un mouvement dans le plan vertical (fonction de réglage d'altitude). Cette combinaison de mouvements dans plusieurs plans fait que les forces de poussées des corps de l'anguille ne sont également plus dans un seul et même plan. Le résultat de cette configuration produit un mouvement que l'on nomme mouvement de roulis et qui se traduit par un mouvement sinusoïdal de la tête de l'anguille autour de son axe \mathbf{x}_0 . Ce mouvement dans le cas de l'anguille est compensé en jouant sur le terme $C_{ad1} = f(C_4)$ du modèle fluide, ce qui a pour effet d'exercer un couple résistant opposé au mouvement de rotation de la tête et du reste du corps (effet stabilisant). Cependant, en utilisant le coefficient de la littérature, on s'aperçoit que la compensation n'est pas suffisante pour contrer le mouvement de roulis. Pour mieux appréhender le rôle de ce paramètre C_4 dans la stabilisation du mouvement de roulis, on effectue une première simulation en utilisant le coefficient de la littérature ($C_4=1$) et une deuxième en prenant un coefficient non réaliste mais qui a pour effet d'augmenter la compensation ($C_4=500$) :

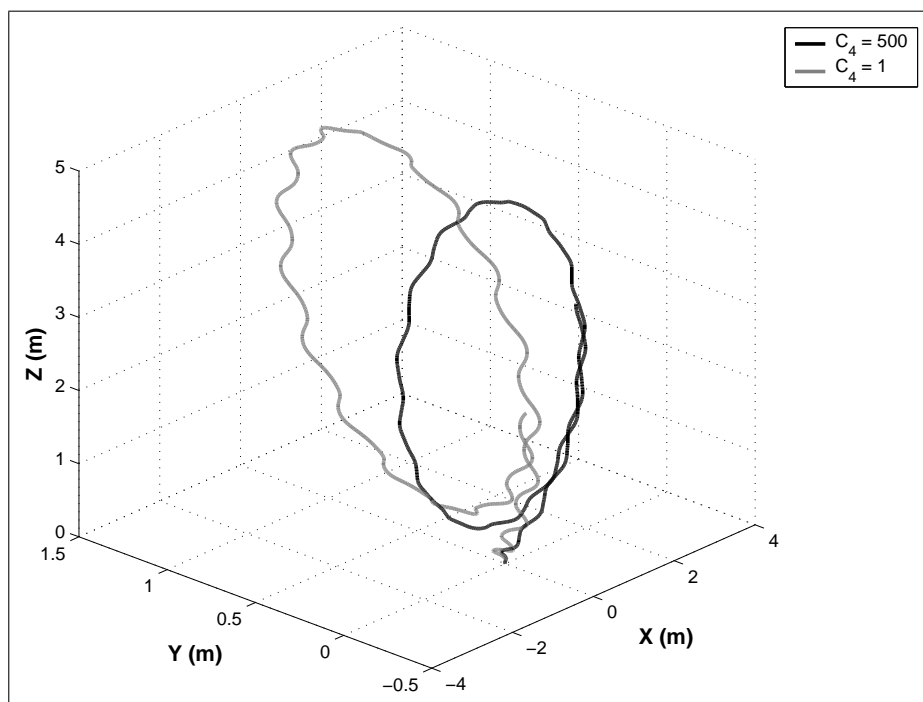


Figure 2.23: – Trajectoire de l'anguille pour différents coefficients du modèle fluide

On voit sur la figure (2.23) qu'en utilisant la valeur numérique de la littérature le résultat diffère de manière significative. Dans les deux cas, l'anguille effectue un looping dans un plan, mais ceux-ci sont décalés l'un par rapport à l'autre. Seule la simulation où $C_4=500$ donne un résultat dont le looping de l'anguille se situe dans le plan $(\mathbf{X}_w, \mathbf{Z}_w)$, ce qui correspond à la trajectoire souhaitée. Par ailleurs, la figure (2.24) montre l'influence du paramètre C_4 sur l'angle de roulis (autour de l'axe \mathbf{x}_0) produit au niveau de la tête de

l'anguille :

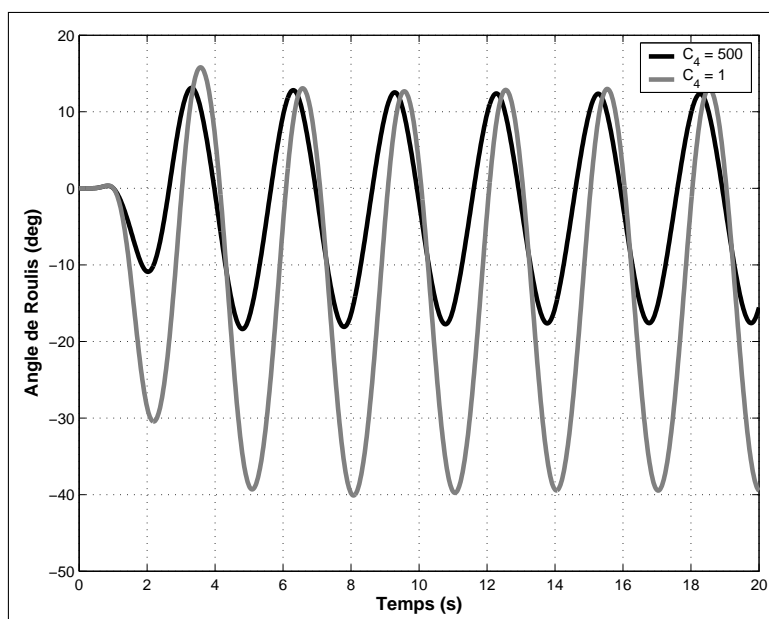


Figure 2.24: – Angle de Roulis de la tête de l'anguille

A partir des figures (2.23) et (2.24), on peut en conclure que l'augmentation du paramètre C_4 permet de diminuer l'amplitude de l'angle de roulis ce qui permet à l'anguille de ne pas dévier pendant la nage. Ceci montre également qu'en apportant un contrôle adéquat du roulis de l'anguille, on peut améliorer la qualité des résultats obtenus. Cependant, les coefficients fluides ne pouvant pas être changés, il faut alors se demander comment une anguille biologique fait pour pouvoir contrôler son angle de roulis pendant des nages 3D. Pour tenter de répondre à cette question, plusieurs explications peuvent être avancées :

- Jusqu'à présent dans nos simulations, les articulations de roulis (articulations 3j) n'ont pas été utilisées pour la nage. On peut alors se demander si l'utilisation de celles-ci peut contribuer à résoudre le problème posé. Seulement, pour l'anguille biologique, ce mouvement n'a qu'un très faible angle de débattement et ne peut donc pas être considéré comme une solution de contrôle si l'on souhaite garder un fonctionnement bio-mimétique.
- Parmi les cas de nage dans un espace 3D, il existe celui des virages inertiels, c'est-à-dire que l'anguille effectue sa manoeuvre en arrêtant l'ondulation du corps. Dans ce cas là, il n'y a donc plus de couplage entre les différents mouvements et le problème du roulis disparaît. Cependant, il ne faut pas oublier que l'eau dissipe l'énergie cinétique de l'anguille et va donc rapidement l'arrêter s'il n'y a aucune ondulation le long du corps. Ce type de nage est donc utile pour faire des manoeuvres courtes sans consommation

d'énergie, mais n'est en aucun cas adapté pour des trajectoires de nage complexes et de durées importantes.

- Le dernier point concerne un élément qui n'a pas été pris en compte dans la modélisation du robot : les nageoires pectorales de l'anguille. A l'instar des sous-marins, l'anguille se sert des nageoires pectorales comme stabilisateurs, mais aussi comme ailerons. En effet, par la présence de ces ailerons, la surface de résistance à l'eau est considérablement augmentée vis-à-vis de la rotation de la tête autour de l'axe de roulis. A noter que la présence des nageoires pectorales peut également aider au réglage en altitude de l'anguille. Elles constituent donc la piste la plus sérieuse à étudier, et encore une fois l'étude du comportement d'une vraie anguille pourrait permettre de mieux comprendre l'utilisation de ces nageoires.

2.10 Modélisation et résultats en ajoutant des ailerons

2.10.1 Modification des algorithmes de Newton-Euler

A partir des résultats présentés précédemment, on a pu remarquer que la présence d'ailerons devenait nécessaire pour atténuer le mouvement de roulis de l'anguille lors de nages tridimensionnelles. Ces deux ailerons qu'on ajoute au robot sont orientables en utilisant deux moteurs intégrés à la tête de l'anguille. La nouvelle modélisation géométrique du robot intégrant la présence des deux nouveaux ailerons est donnée figure (2.25) :

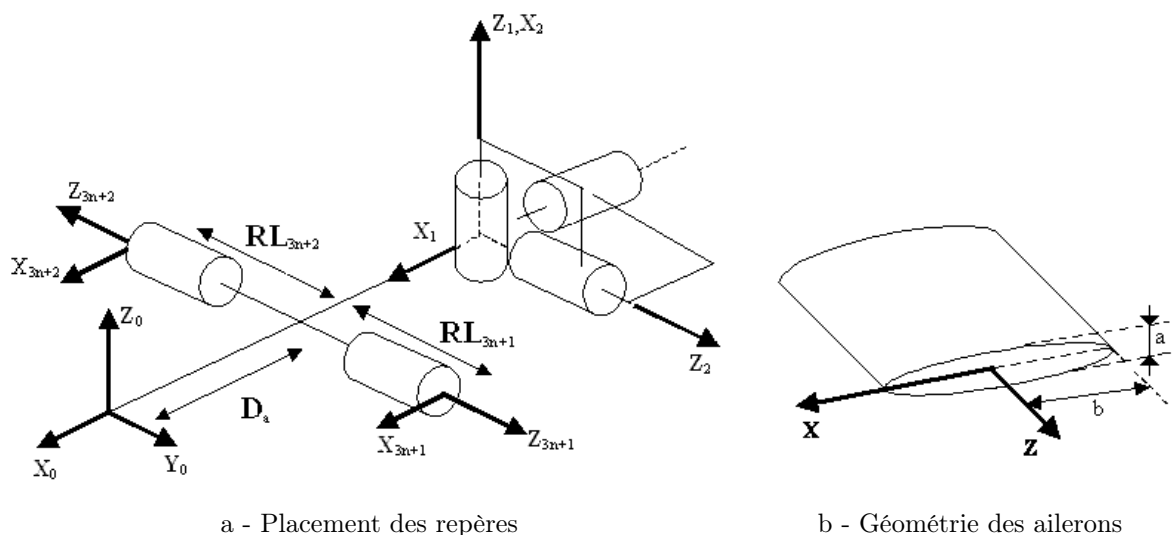


Figure 2.25: – Modélisation géométrique du robot sériel avec les ailerons

Avec l'ajout de ces deux ailerons, le robot n'a plus une structure sérielle, mais une structure arborescente puisque trois corps sont à présent fixés au corps de tête. A partir de cette

modélisation, on peut donc en déduire le nouveau tableau des paramètres géométriques :

j	a(j)	α_j	d_j	θ_j	r_j	q_{0j}
1	0	0	D_1	θ_1	0	0
2	1	$-\pi/2$	0	θ_2	0	$\pi/2$
3	2	$-\pi/2$	0	θ_3	RL_3	$\pi/2$
4	3	$\pi/2$	0	θ_4	0	$-\pi/2$
...						
3n-1	3n-2	$-\pi/2$	0	θ_{3n-1}	0	$-\pi/2$
3n	3n-1	$-\pi/2$	0	θ_{3n}	RL_{3n}	$\pi/2$
3n+1	0	$-\pi/2$	D_a	θ_{3n+1}	RL_{3n+1}	0
3n+2	0	$\pi/2$	D_a	θ_{3n+2}	RL_{3n+2}	0

Tableau 2.3: – Paramètres géométriques de la structure arborescente

Comme pour tout robot à structure arborescente, une nouvelle colonne fait son apparition dans le tableau des paramètres géométriques. La colonne a(j) désigne ici le "corps antécédent", c'est-à-dire le numéro du corps auquel le corps j est relié. On remarque dans le cas présent que 3 corps ont pour antécédent le corps 0 : les deux ailerons et le premier corps de la colonne vertébrale de l'anguille.

Comme l'ajout des ailerons est prévu pour changer le comportement dynamique de l'anguille, il faut en conséquence modifier le modèle dynamique inverse pour en tenir compte. Vu que ces modifications ne concernent que le corps de tête de l'anguille, il suffit donc de modifier les équations (2.22) et (2.23) dans le cas où j=0. Cela revient donc à écrire :

$${}^0\mathbb{J}_0^{c,a} = {}^0\mathbb{J}_0^c + \sum_{k/a(k)=0, k \neq 1} {}^k\mathbb{T}_0^T {}^k\mathbb{J}_k {}^k\mathbb{T}_0 \quad (2.90)$$

$${}^0\beta_0^{c,a} = {}^0\beta_0^c + \sum_{k/a(k)=0, k \neq 1} {}^k\mathbb{T}_0^T ({}^k\beta_k + {}^k\mathbb{J}_k {}^k\gamma_k) \quad (2.91)$$

On peut ensuite calculer l'accélération de la tête avec l'équation (2.24) modifiée :

$${}^0\dot{\mathbb{V}}_0 = - ({}^0\mathbb{J}_0^{c,a})^{-1} {}^0\beta_0^{c,a} \quad (2.92)$$

Pour les simulations qui vont suivre, nous avons choisit d'utiliser des ailerons dont les caractéristiques sont les suivantes :

- longueur de l'aileron : 0,2 m ;
- section elliptique dont les demi-petit et demi-grand axe sont respectivement :

$$a=0,01 \text{ m } b=0,06 \text{ m}$$

- masse : 377 g ;
- moment d'inertie :

$$\mathbf{MS} = \begin{bmatrix} 0 & 0 & 0.0377 \end{bmatrix}^T$$

- matrice d'inertie :

$$\mathbf{J} = \begin{bmatrix} 5.0360e^{-3} & 0 & 0 \\ 0 & 5.3658e^{-3} & 0 \\ 0 & 0 & 3.4872e^{-4} \end{bmatrix}$$

- distance des ailerons par rapport à la tête : 0,3m
- distance des ailerons par rapport à la colonne vertébrale : $RL_{3n+1} = RL_{3n+2} = 0,065$ m

2.10.2 Loi de contrôle pour régler le roulis par les ailerons

L'ajout des ailerons sur le robot permet de produire plusieurs types de mouvements. Si on applique des consignes de même signe pour les articulations q_{3n+1} et q_{3n+2} , cela va produire une rotation de l'anguille autour du vecteur \mathbf{x}_0 , c'est à dire un mouvement selon l'axe de roulis. Par contre, si on applique des consignes de signes opposées pour ces deux articulations, on va permettre à l'anguille de monter ou de plonger.

De ces constatations, on peut écrire deux types de couplages pour les ailerons :

- un couplage en différentiel q_d qui permet d'agir sur le roulis de la tête ;
- un couplage en additionnel q_a qui permet d'agir sur la profondeur de nage.

Ainsi, le fonctionnement que l'on vient de voir peut se traduire par les équations suivantes :

$$q_{3n+1} = q_d - q_a \quad (2.93)$$

$$q_{3n+2} = q_d + q_a \quad (2.94)$$

avec :

- $q_d < 0$ pour avoir un mouvement de rotation positif autour de l'axe de la tête \mathbf{x}_0 ;
- $q_a > 0$ pour faire monter l'anguille.

Comme dans le cas présent, seul le contrôle du roulis nous intéresse, on posera $q_a = 0$.

Pour contrôler l'angle de roulis afin que celui-ci soit asservi sur un angle de consigne ϕ_d , nous proposons d'utiliser la loi de commande suivante :

$$q_d = \beta_{\max} \tanh(k_1(\phi - \phi_d) - k_2 \omega_{x,0}) \quad (2.95)$$

où :

- β_{\max} représente l'angle maximum des ailerons ;
- ϕ représente l'angle de roulis mesuré ;
- $\omega_{x,0}$ représente la composante suivant l'axe x du vecteur ${}^0\omega_0$;
- k_1 et k_2 sont des coefficients de réglage.

On remarque que l'utilisation de la fonction tangente hyperbolique permet d'obtenir un effet de saturation naturel de cette loi de commande.

Pour tester cette loi de commande, nous allons reprendre la simulation de la nage 3D en looping afin de pouvoir comparer les résultats que nous avons obtenus précédemment (sans utiliser les ailerons) avec ceux obtenus en utilisant les ailerons et la loi de commande proposée. Pour ces simulations, les valeurs numériques des paramètres utilisés dans le générateur de mouvement sont les suivants :

$$A=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; t_f=3 \text{ s} ; \lambda=1,5 \text{ m} ; T=3 \text{ s} ; O_{ff_1}=0 ; O_{ff_2}=\frac{\pi}{40} ; \\ \phi_d = 0 ; \beta = 30^\circ ; k_1=4 ; k_2=8$$

Nous avons vu lors des simulations précédentes que pour obtenir une trajectoire convenable il fallait augmenter le paramètre C_4 du modèle fluide ($C_4=500$ dans notre simulation), mais que cela n'était pas réaliste par rapport à la valeur correspondant au monde réel ($C_4=1$). Pour pouvoir quantifier l'apport des ailerons dans le calcul du modèle dynamique de la nage, nous allons réaliser ces deux simulations :

- une simulation sans les ailerons et pour laquelle on prend $C_4=500$ et qui correspond au fonctionnement "idéal" (bien que non réaliste) que l'on souhaite obtenir.
- une simulation en utilisant, les ailerons, la loi de commande proposée et pour laquelle $C_4=1$, correspondant ainsi au cas réel.

Les figures (2.26 -a) et (2.26 -b) représentent la trajectoire de l'anguille au cours du temps et respectivement représentées dans les plans \mathbf{x}_w - \mathbf{y}_w et \mathbf{y}_w - \mathbf{z}_w du repère fixe Σ_w :

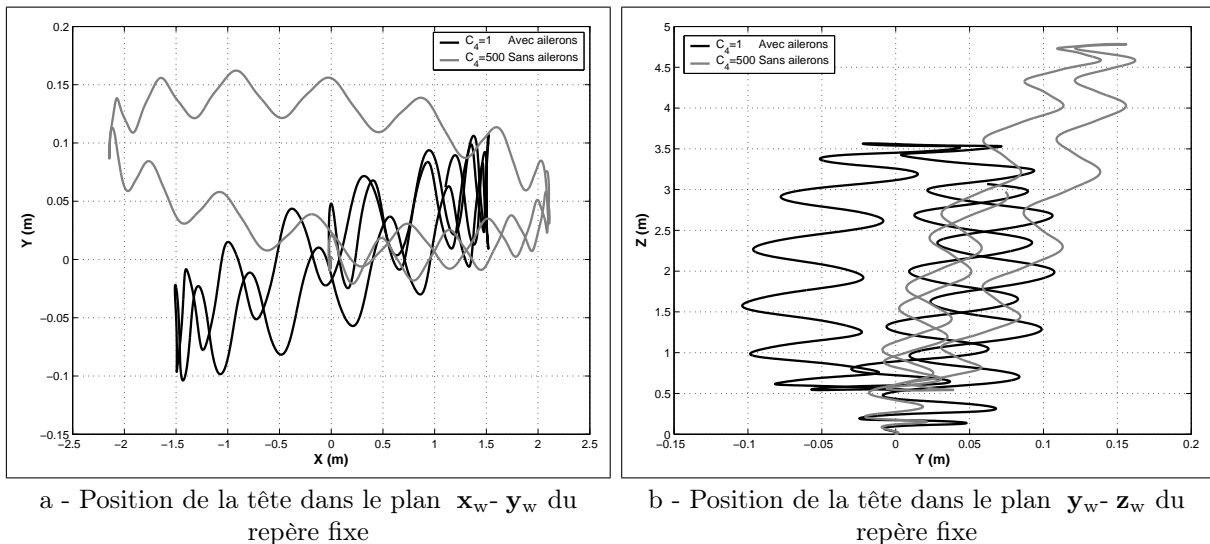


Figure 2.26: – Comparaison des trajectoires de l'anguille avec et sans ailerons

L'analyse de ces deux figures est la suivante :

Simulation sans les ailerons :

Sans les ailerons, on peut voir sur la figure (2.26 -b) qu'on obtient une trajectoire dans un plan vertical ayant légèrement tourné autour de l'axe \mathbf{x}_w . Cette rotation du plan se caractérise sur la figure (2.26 -a) par un tracé de la trajectoire ayant une forme elliptique.

Simulation avec les ailerons et la loi de commande :

En utilisant les ailerons, on observe cette fois-ci sur la figure (2.26 -a) que la trajectoire se situe bien dans un plan vertical, mais que celle-ci est légèrement déviée par rapport à l'axe des x. Cette déviation du plan se traduit sur la figure (2.26 -b) par un tracé en forme d'ellipse. On peut également rappeler que ce genre de déviation a été obtenu lors de la simulation en nage plane, et que ceci est dû au transitoire dans l'établissement de la loi d'ondulation ce qui provoque des efforts latéraux dissymétriques.

A la fin de cette analyse, on peut conclure que l'utilisation des ailerons permet d'obtenir les caractéristiques souhaitées de la nage et sans avoir à utiliser de coefficients irréalistes pour le modèle fluide .

Pour la nage 3D, la caractéristique principale qui nous intéresse est le fait de pouvoir nager dans un plan vertical (et pas légèrement penché comme dans le cas de la simulation sans ailerons). Pour pouvoir caractériser si le plan de nage est bien vertical ou alors penché, il suffit d'analyser l'angle de roulis de la tête au cours du temps. Pour cela, les figures (2.27 -a) et (2.27 -b) représentent respectivement l'angle de roulis obtenu pour les deux simulations et la moyenne de cet angle au cours du temps :

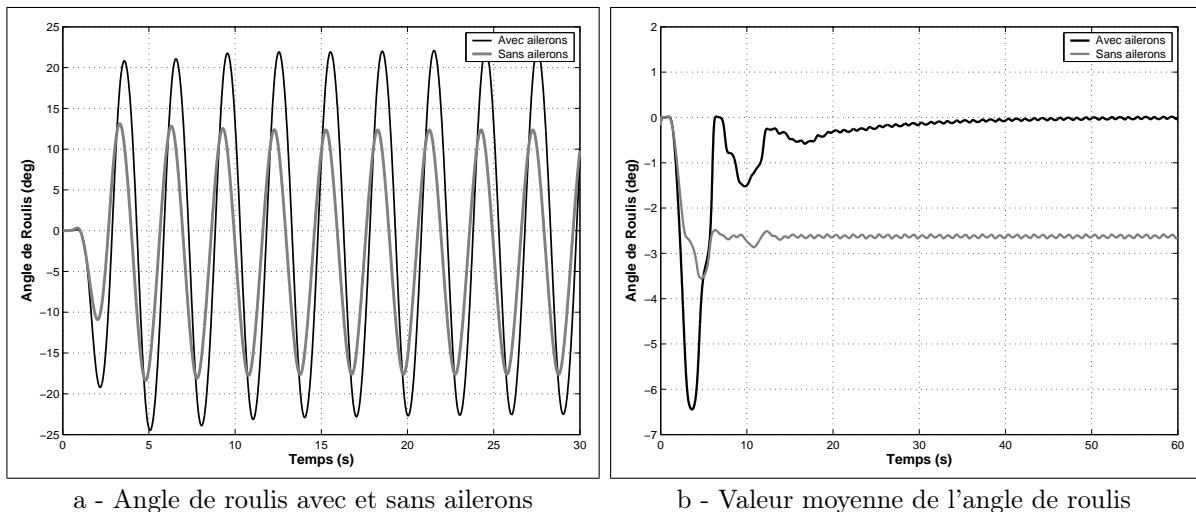


Figure 2.27: – Angle de roulis de l'anguille durant une nage 3D et avec ou sans ailerons

Contrairement à ce que l'on pouvait s'attendre, la figure (2.27 -a) nous montre que l'utilisation des ailerons ne réduit pas de manière significative l'amplitude du mouvement de

roulis. Cependant, lorsqu'on regarde sur la figure (2.27-b), on s'aperçoit que la valeur moyenne n'est pas la même pour les deux simulations. Dans le cas de la simulation sans les ailerons, la valeur moyenne de l'angle de roulis se stabilise sur une valeur non nulle (-2.7° environ) ce qui explique que le plan dans lequel l'anguille évolue soit penché par rapport à la verticale. En revanche, avec les ailerons et la loi de commande utilisée, cela nous permet d'asservir la moyenne de l'angle de roulis sur la consigne ϕ_d (ici égal à 0) et permet donc d'obtenir une trajectoire se situant dans un plan bien vertical.

2.11 Conclusion

Ce chapitre présente les modèles dynamiques inverse et direct du robot anguille sous forme sérielle. L'algorithme proposé permet d'obtenir un modèle dynamique 3D qui peut être facilement utilisé en simulation. Les modèles dynamiques sont développés en utilisant le formalisme récursif de Newton-Euler, de telle manière que le modèle inverse puisse calculer les accélérations de la tête et les couples moteurs des articulations, tandis que le modèle direct permet de calculer les accélérations de la tête et des articulations. Ces algorithmes constituent une généralisation de ceux utilisés pour les robots manipulateurs. La généralisation prend en compte la mobilité de cette base, qui constitue dans notre cas la tête de l'anguille. De plus, en se basant sur la littérature de la mécanique des fluides, un modèle simplifié du contact fluide-structure a été utilisé. L'algorithme proposé ici peut être facilement utilisé pour d'autres types de systèmes (robots serpents, robots volants,...) en prenant un modèle approprié des forces extérieures.

Cette première structure de robot a permis d'avoir un ordre de grandeur permettant de dimensionner les moteurs qui sont utilisés sur le prototype. Ce même dimensionnement nous montre que sur les 3 moteurs utilisés dans chacune des rotules, les moteurs de lacet (articulations 3j-2) nécessitent beaucoup plus de couple comparé aux deux autres. Cette constatation permet de s'interroger sur le choix d'une telle structure mécanique et nous a ainsi conduit à l'élaboration du robot présenté dans le chapitre suivant qui permet de mieux répartir les couples nécessaires.

Enfin, les simulations ont permis de montrer qu'il était possible de faire nager l'anguille dans un espace tridimensionnel en utilisant des lois de mouvements relativement simples. Nous avons également pu montrer l'intérêt de rajouter des ailerons et une loi de commande simple pour pouvoir contrôler le mouvement de roulis et obtenir des nages 3D réalistes.

Chapitre 3

Modélisation d'un module de l'anguille

3.1 Introduction

Dans ce chapitre est présenté la modélisation d'une structure parallèle réalisée dans le cadre du projet robot anguille. Le robot étant conçu dans une approche bio-mimétique, nous avons cherché à avoir un comportement le plus proche possible de l'être vivant [Chablat et Wenger, 2005]. Dans le chapitre 2, nous avons vu que la colonne vertébrale de l'anguille pouvait être assimilée à une succession de liaisons rotules. Pour la réalisation des articulations rotules entre vertèbres, le choix d'une structure de type parallèle s'est révélé être le meilleur car il permet d'allier la compacité, la robustesse ainsi que l'augmentation et la répartition des efforts pouvant être transmis. Le faible espace de travail qui accompagne en général ce type de robot ne constitue pas un handicap pour notre étude car les débattements souhaités (en tant que vertèbre de l'anguille) ne sont pas trop importants. Pour pouvoir actionner les différents degrés de liberté de la liaison rotule, nous nous sommes encore une fois inspirés de la nature en utilisant le principe des muscles extenseurs et fléchisseurs que l'on retrouve dans les êtres vivants (figure 3.1).

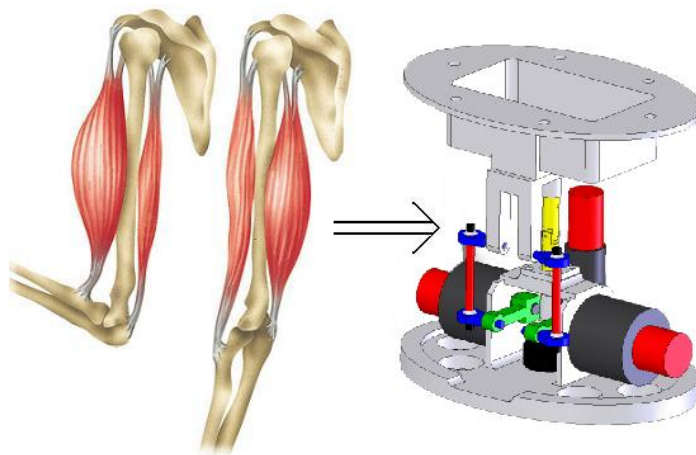


Figure 3.1: – Muscles extenseurs et fléchisseurs permettant de faire bouger le squelette

D'après les observations menées sur les anguilles, seules les vertèbres permettant la propulsion de l'anguille sont actionnées et possèdent des débattements importants, les autres étant passives avec de faibles débattements. Pour notre robot, nous avons décidé d'actionner l'ensemble des degrés de liberté afin d'augmenter la manoeuvrabilité globale. Cependant, pour se rapprocher du comportement de l'anguille, on se limite dans cette étude à des angles de $\pm 15^\circ$ pour l'angle de tangage, $\pm 30^\circ$ pour l'angle de lacet et $\pm 4^\circ$ pour l'angle de roulis.

Ces différentes considérations ont donc menés à la structure mécanique de la figure (3.2), réalisée par l'équipe de Méthode de Conception Mécanique de l'IRCCyN¹ :

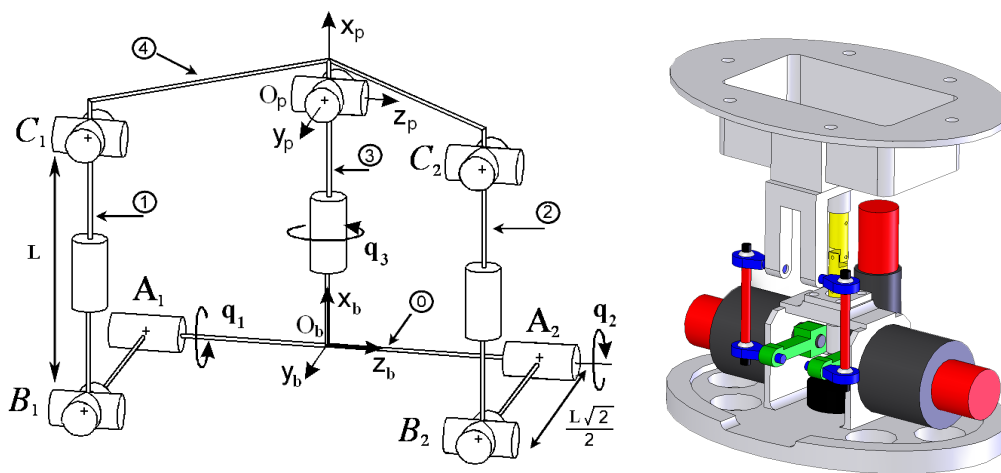


Figure 3.2: – Structure et modèle CAO du module parallèle

Ce module parallèle est composé d'une base (notée 0), de 3 jambes ①, ②, ③ et d'une plate-forme ④. Le module est composé d'une structure R(RRR)(RR) pour les jambes ① et ② et d'une structure R(RR) pour la jambe ③, où (RR) désigne une articulation de type cardan et (RRR) une articulation de type rotule. Chaque jambe possède un seul actionneur placé sur la base, tandis que la plate-forme contient les circuits électroniques qui servent à la commande des moteurs.

En faisant l'analogie avec un être vivant, la base et la plate-forme de ce module constituent 2 os qui se suivent (le mouvement de rotule entre les 2 se situe au point O_p). Les 2 muscles antagonistes sont quant à eux représentés par les 2 jambes ① et ②. Contrairement à la structure série étudiée dans le chapitre 2 où chaque moteur permettait d'actionner un seul mouvement, on remarque dans cette nouvelle structure que les moteurs des jambes ① et ② fonctionnent ensemble pour produire le mouvement de nage, ce qui permet d'avoir

¹Institut de Recherche en Communications et Cybernétique de Nantes

une meilleure répartition des efforts.

Pour la modélisation de ce module, nous avons choisi comme variable d'état le vecteur Θ des angles de Roulis-Tangage-lacet (respectivement définis comme les angles autour des axes \mathbf{x} , \mathbf{y} et \mathbf{z}). Or, les variables de commande étant les articulations motorisées q_i de ce module, on propose d'utiliser le schéma de commande suivant (figure 3.3) :

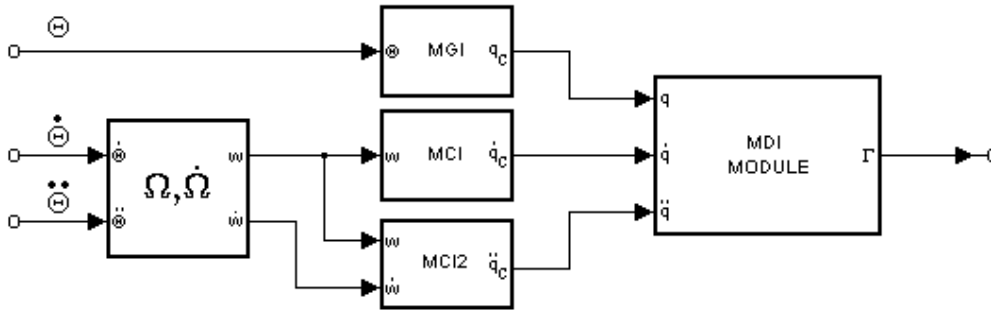


Figure 3.3: – Schéma de commande

A partir de notre vecteur d'état Θ et de ses dérivés, on commence par calculer la vitesse et l'accélération ($\mathbf{w}, \dot{\mathbf{w}}$) de la plate-forme par rapport à la base du module, puis en utilisant les modèles géométrique et cinématique inverses, on détermine les positions, vitesses et accélérations des articulations motorisées. Le calcul des couples articulaires Γ s'obtient finalement en utilisant le modèle dynamique inverse du module. Dans la suite, on fera l'hypothèse que les articulations motorisées \mathbf{q} suivent parfaitement les consignes articulaires \mathbf{q}_c .

Ce chapitre est organisé de la manière suivante : dans les paragraphes 3.2 à 3.4 sont présentés le modèle géométrique inverse et les modèles cinématiques du module. Ensuite, dans le paragraphe 3.5 est présenté le modèle dynamique inverse du module ainsi qu'une hypothèse permettant de simplifier le calcul de celui-ci. Pour finir, le paragraphe 3.6 permet de déterminer les limites du module à partir du calcul de l'espace de travail.

3.2 Modèle géométrique inverse du module

Le modèle géométrique permet de calculer les positions articulaires motorisées en fonction de la situation (ici orientation) de la plate-forme de chaque module. Pour cela, on commence par affecter un repère Σ_p de centre O_p lié à la plate-forme et un repère Σ_b lié à la base du module et dont le centre O_b se situe à une distance d de O_p suivant l'axe \mathbf{x}_b . Le placement de ces deux repères est donné sur la figure (3.4) :

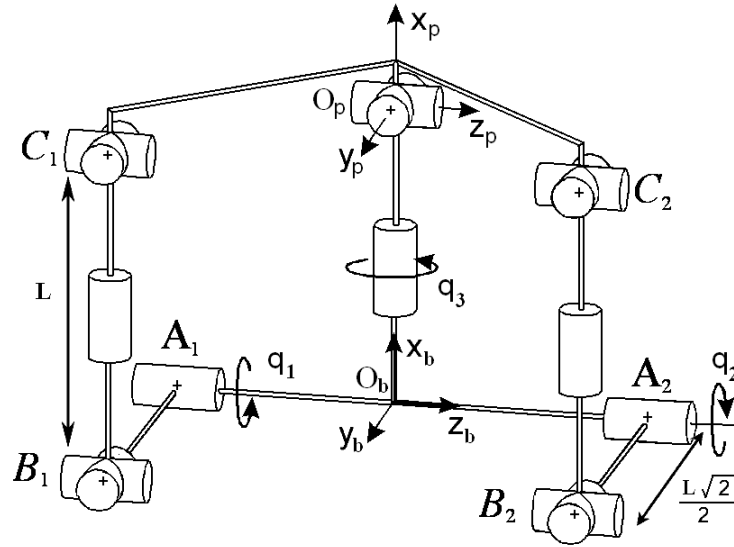


Figure 3.4: – Placement des repères sur le module

Nous avons choisi de paramétrer la rotation de la plate-forme du module par rapport à la base par les angles de Roulis-Tangage-Lacet. La matrice de transformation entre les repères Σ_p et Σ_b sera donc exprimée en fonction des paramètres suivants :

- $\Theta = [\theta \ \phi \ \psi]^T$ les 3 angles de cardan ;
- d : la distance entre les axes z_b et z_p le long de l'axe x_b .

La matrice de transformation homogène peut donc s'écrire de cette manière :

$${}^bT_p = \text{Trans}(x, d) \text{Rot}(x, \theta) \text{Rot}(y, \phi) \text{Rot}(z, \psi) =$$

$$\begin{bmatrix} C\phi C\psi & -C\phi S\psi & S\phi & d \\ S\theta S\phi C\psi + C\theta S\psi & -S\theta S\phi S\psi + C\theta C\psi & -S\theta C\phi & 0 \\ -C\theta S\phi C\psi + S\theta S\psi & C\theta S\phi S\psi + S\theta C\psi & C\theta C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Dans la suite, l'exposant situé en haut à gauche indique le repère de projection et la matrice bT_p est décomposée de la manière suivante :

$${}^bT_p = \begin{bmatrix} {}^bR_p & {}^bP_p \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

avec :

- bR_p la matrice d'orientation (3×3) du repère Σ_p par rapport au repère Σ_b ,
- bP_p le vecteur position (3×1) du repère Σ_p par rapport au repère Σ_b .

Tableau des paramètres géométriques

Pour pouvoir établir le tableau des paramètres géométriques du robot, on place pour les 3 jambes du module les repères en utilisant les techniques utilisées pour les structures arborescentes [Khalil et Dombre, 2002]. On ne tiendra pas compte ici de la plate-forme que l'on déconnecte complètement :

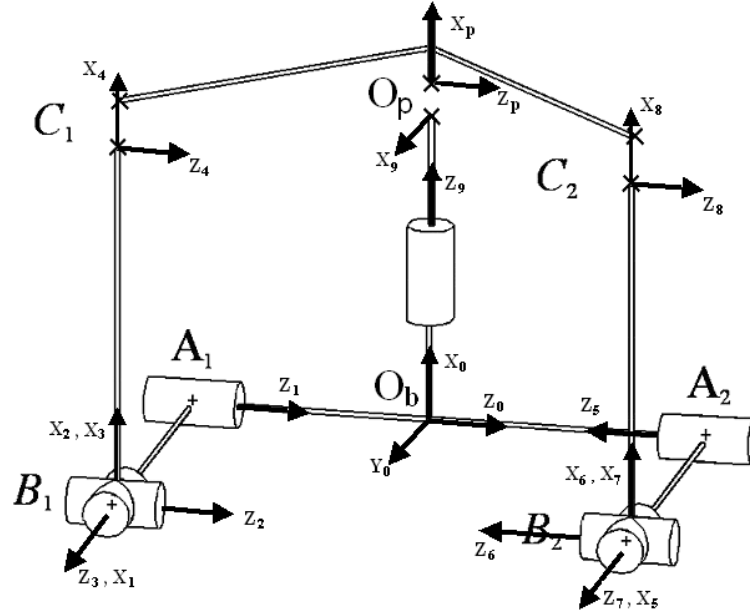


Figure 3.5: – Placement des repères pour le tableau des paramètres géométriques

Avec cette description géométrique, on en déduit que les jambes 1 à 3 sont respectivement composées des corps C_1 à C_4 , C_5 à C_8 et C_9 .

A partir de la figure (3.5), il est possible d'écrire le tableau des paramètres géométriques. On définit $i=a(j)$ comme le corps antécédent au corps j et on envisage deux cas de figures :

1. si le vecteur \mathbf{x}_i est perpendiculaire à \mathbf{z}_i et \mathbf{z}_j : on définit alors $\mathbf{u}_j = \mathbf{x}_j$;
2. si le vecteur \mathbf{x}_i est perpendiculaire à \mathbf{z}_i et un autre \mathbf{z}_k , alors on définit \mathbf{u}_j comme étant la perpendiculaire commune à \mathbf{z}_i et \mathbf{z}_j .

Le tableau des paramètres s'écrit alors en fonction des 6 paramètres suivants :

- γ_j : l'angle entre \mathbf{x}_i et \mathbf{u}_j autour de \mathbf{z}_i ;
- b_j : la distance entre \mathbf{x}_i et \mathbf{u}_j e long de \mathbf{z}_i ;

Et les quatre paramètres usuels :

- α_j : l'angle entre \mathbf{z}_i et \mathbf{z}_j autour de \mathbf{u}_j ,
- d_j : la distance entre \mathbf{z}_i et \mathbf{z}_j selon \mathbf{u}_j ,
- θ_j : l'angle entre \mathbf{u}_j et \mathbf{x}_j autour de \mathbf{z}_j ,

- r_j : la distance entre \mathbf{u}_j et \mathbf{x}_j selon \mathbf{z}_j .

j	a(j)	μ_j	σ_j	α_j	d _j	q _j	r _j	γ_j	b _j
1	0	1	0	0	0	q ₁	-r ₁	0	0
2	1	0	0	0	d ₂	q _{1,2}	0	0	0
3	2	0	0	$-\pi/2$	0	q _{1,3}	0	0	0
4	3	0	2	$\pi/2$	d ₄	0	0	0	0
5	0	1	0	π	0	q ₂	-r ₅	0	0
6	5	0	0	0	d ₆	q _{2,2}	0	0	0
7	6	0	0	$\pi/2$	0	q _{2,3}	0	0	0
8	7	0	2	$\pi/2$	d ₈	0	0	0	0
9	0	1	0	$\pi/2$	0	q ₃	r ₉	$\pi/2$	0

Tableau 3.1: – Tableau des paramètres géométriques

avec :

- $r_1 = r_5 = \|\mathbf{O}_b \mathbf{A}_i\| = L \frac{\sqrt{2}}{2}$;
- $d_2 = d_6 = \|\mathbf{A}_i \mathbf{B}_i\| = L \frac{\sqrt{2}}{2}$;
- $d_4 = d_8 = \|\mathbf{B}_i \mathbf{C}_i\| = L$;
- $r_9 = \|\mathbf{O}_b \mathbf{O}_p\| = L$.

Nous avons choisi d'introduire un rapport de $\frac{\sqrt{2}}{2}$ entre les longueurs $\|\mathbf{A}_i \mathbf{B}_i\|$ et $\|\mathbf{B}_i \mathbf{C}_i\|$ car cela permet de simplifier le calcul des modèles géométriques.

Dans le cas du modèle géométrique inverse, le fait de ne pas considérer la plate-forme dans la modélisation du robot permet également de simplifier les calculs : connaissant la posture (ou la matrice d'orientation) de la plate-forme, il suffit d'imposer des contraintes géométriques aux organes terminaux des 3 jambes pour résoudre le modèle géométrique au complet. Ces contraintes géométriques sont :

- une contrainte en position au niveau des points \mathbf{C}_i pour les jambes 1 et 2 (l'articulation entre l'organe terminal de la jambe et la plate-forme étant une rotule).
- une contrainte en rotation au point \mathbf{O} pour la jambe 3 (l'articulation entre l'organe terminal de la jambe et la plate-forme étant un cardan).

Le calcul du modèle géométrique inverse du module peut être ramené à l'étude des modèles géométriques de chacune des 3 jambes du module :

Modèle géométrique inverse de la jambe 3

La jambe 3 est le cas le plus simple à traiter. A partir de l'expression de la matrice de transformation (équation 3.1), on remarque que la première rotation se fait autour de l'axe \mathbf{x}_b , c'est-à-dire l'axe de l'articulation motorisée q_3 . Ainsi on peut en déduire immédiatement que :

$$q_3 = \theta \quad (3.2)$$

Modèle géométrique inverse des jambes 1 et 2

Le calcul des positions articulaires se fait en trois étapes. Premièrement, on calcule la position des points C_i (pour $i=1,2$) dans le repère de la base Σ_b :

$${}^b\mathbf{P}_{C_i} = {}^b\mathbf{R}_p {}^p\mathbf{P}_{C_i} + {}^b\mathbf{P}_p \quad (3.3)$$

$${}^p\mathbf{P}_{C_1} = L \frac{\sqrt{2}}{2} \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}^T \quad (3.4)$$

$${}^p\mathbf{P}_{C_2} = L \frac{\sqrt{2}}{2} \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T \quad (3.5)$$

$${}^b\mathbf{P}_p = \begin{bmatrix} L & 0 & 0 \end{bmatrix}^T \quad (3.6)$$

avec L représentant la longueur du vecteur $\overrightarrow{B_i C_i}$.

Dans un second temps, on peut calculer la position des points B_i (pour $i=1,2$) en fonction de la position articulaire motorisée q_i dans le repère de la base Σ_b :

$${}^b\mathbf{B}_1 = \begin{bmatrix} -S_{q_1} L \frac{\sqrt{2}}{2} & C_{q_1} L \frac{\sqrt{2}}{2} & -L \frac{\sqrt{2}}{2} \end{bmatrix}^T \quad (3.7)$$

$${}^b\mathbf{B}_2 = \begin{bmatrix} S_{q_2} L \frac{\sqrt{2}}{2} & C_{q_2} L \frac{\sqrt{2}}{2} & L \frac{\sqrt{2}}{2} \end{bmatrix}^T \quad (3.8)$$

où $C_{q_i} = \cos(q_i)$, $S_{q_i} = \sin(q_i)$.

Finalement, la troisième étape consiste à écrire la contrainte géométrique en distance qui lie les points B_i et C_i , soit $\|\mathbf{B}_i \mathbf{C}_i\| = L$. Pour simplifier les calculs, on pose également les deux équivalences suivantes :

$$\begin{aligned} S_{q_i} &= \frac{2Q_i}{1 + Q_i^2} \\ C_{q_i} &= \frac{1 - Q_i^2}{1 + Q_i^2} \end{aligned}$$

où $Q_i = \tan(q_i/2)$.

Le modèle géométrique inverse se résume alors à trouver la solution d'une équation polynomiale du second degré telle que (pour $i=1,2$) :

$$\alpha_{i1}Q_i^2 + \alpha_{i2}Q_i + \alpha_{i3} = 0 \quad (3.9)$$

avec :

$$\alpha_{i1} = \frac{\sqrt{2}}{L} \left(L\sqrt{2} + P_{Ciy} + e_i P_{Ciz} + \sqrt{2}P_{Cix} \right) \quad (3.10)$$

$$\alpha_{i2} = 2\sqrt{2}e_i \left(1 + \frac{P_{Cix}}{L} \right) \quad (3.11)$$

$$\alpha_{i3} = \frac{\sqrt{2}}{L} \left(L\sqrt{2} - P_{Ciy} + e_i P_{Ciz} + \sqrt{2}P_{Cix} \right) \quad (3.12)$$

$${}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci} = \begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T \quad (3.13)$$

$$e_1 = 1 \quad \text{et} \quad e_2 = -1 \quad (3.14)$$

Cette équation du second degré permet d'obtenir deux solutions qui correspondent aux deux configurations "Coude haut" et "Coude bas". Pour la jambe i ($i=1,2$), les deux solutions sont :

$$Q_{i,1} = \frac{-\alpha_{i2} + \sqrt{\alpha_{i2}^2 - 4\alpha_{i1}\alpha_{i3}}}{2\alpha_{i1}} \quad \text{ou} \quad Q_{i,2} = \frac{-\alpha_{i2} - \sqrt{\alpha_{i2}^2 - 4\alpha_{i1}\alpha_{i3}}}{2\alpha_{i1}} \quad (3.15)$$

En tenant compte de la géométrie des deux jambes et des pièces mécaniques utilisées, elles ne peuvent pas, dans leur domaine de fonctionnement, changer de configuration : une seule solution est donc valable sur le domaine d'étude. Pour savoir laquelle des deux solutions est la bonne, on prend le cas simple où la plate-forme du module est parallèle à la base soit $\theta = \phi = \psi = 0$ et $q_1 = q_2 = 0$ (la matrice de rotation devient donc la matrice identité). Les solutions de l'équation deviennent alors :

$$Q_{1,1} = 0 \quad \text{ou} \quad Q_{1,2} = -\sqrt{2} \quad \text{et} \quad Q_{2,1} = -\sqrt{2} \quad \text{ou} \quad Q_{2,2} = 0$$

On voit que seule la solution $Q_{i,j} = 0$ est correcte ce qui signifie que la solution du modèle géométrique est la suivante (pour $i=1,2$) :

$$Q_i = \frac{-\alpha_{i2} + e_i \sqrt{\alpha_{i2}^2 - 4\alpha_{i1}\alpha_{i3}}}{2\alpha_{i1}} \quad (3.16)$$

$$e_1 = 1 \quad \text{et} \quad e_2 = -1$$

Puis :

$$q_i = 2(\text{atan}(Q_i)) \quad (3.17)$$

Simplification des résultats

On a pu voir dans les différents calculs que le modèle géométrique inverse dépend de la taille du mécanisme (via la longueur L). En remarquant que pour les différentes équations écrites précédemment, il est possible de les factoriser par $(L\frac{\sqrt{2}}{2})$, et grâce au fait que $\| \mathbf{A}_i \mathbf{B}_i \| = \frac{\sqrt{2}}{2} \| \mathbf{B}_i \mathbf{C}_i \|$, cela permet de réécrire les équations du second degré (équations (3.10) à (3.12)) sans que les dimensions du module n'apparaissent. On obtient alors :

$$\alpha_{i1} = 2 + P_{Ciy} + e_i P_{Ciz} + \sqrt{2} P_{Cix} \quad (3.18)$$

$$\alpha_{i2} = 2e_i \left(\sqrt{2} + P_{Cix} \right) \quad (3.19)$$

$$\alpha_{i3} = 2 - P_{Ciy} + e_i P_{Ciz} + \sqrt{2} P_{Cix} \quad (3.20)$$

$$e_1 = 1 \quad \text{et} \quad e_2 = -1 \quad (3.21)$$

avec :

$$\begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci} \quad \text{et} \quad {}^p\mathbf{P}_{Ci} = \begin{bmatrix} 0 & 1 & -e_i \end{bmatrix}^T$$

Cette simplification est donc intéressante car elle évite d'introduire des termes de longueurs dans les calculs. Cependant, cela implique d'avoir une bonne précision lors de la réalisation et du montage, afin de garder les bonnes proportionnalités.

3.3 Modèles cinématiques du module

3.3.1 Définitions

Le modèle cinématique direct du module permet de calculer la vitesse angulaire de la plate-forme en fonction des variables articulaires motorisées. Ce modèle s'exprime de manière classique par le biais de la matrice jacobienne selon la relation :

$$\mathbf{w}_p = \mathbf{J}_p \dot{\mathbf{q}}_a \quad (3.22)$$

où :

- \mathbf{w}_p : Vitesse angulaire relative de la plate-forme par rapport à la base du module ;
- \mathbf{J}_p : Matrice (3×3) jacobienne du module ;
- $\dot{\mathbf{q}}_a$: Vecteur des vitesses articulaires motorisées du module :

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}$$

Dans notre cas, il faut pouvoir exprimer les vitesses des actionneurs en fonction de la vitesse de consigne (vitesse de la plate-forme). Pour cela, il suffit d'inverser l'équation (3.22) :

$$\dot{\mathbf{q}}_a = \mathbf{J}_p^{-1} \mathbf{w}_p \quad (3.23)$$

avec \mathbf{J}_p^{-1} la matrice jacobienne inverse.

3.3.2 Modèle cinématique opérationnel

Le modèle cinématique opérationnel inverse permet de calculer les vitesses articulaires $\dot{\mathbf{q}}_a$ en fonction de la vitesse des coordonnées opérationnelles $\dot{\mathbf{X}}$:

$$\dot{\mathbf{q}}_a = \mathbf{J}_x^{-1} \dot{\mathbf{X}} \quad (3.24)$$

Avec \mathbf{X} une représentation de la situation de la plate-forme et qui correspond dans notre étude au vecteur d'état Θ des angles de roulis-tangage-lacet.

Le passage du modèle cinématique classique au modèle cinématique opérationnel se fait en exprimant la vitesse de la plate-forme \mathbf{w}_p en fonction du vecteur d'état Θ :

$$\mathbf{w}_p = \Omega_{RTL} \dot{\Theta} \quad (3.25)$$

avec : Ω_{RTL} une matrice (3×3) de projection.

On pourra calculer \mathbf{w}_p dans le repère Σ_p ou Σ_b en utilisant respectivement dans la relation (3.25) les matrices ${}^p\Omega_{RTL}$ ou ${}^b\Omega_{RTL}$ définies par :

$${}^b\Omega_{RTL} = \begin{bmatrix} 1 & 0 & S\phi \\ 0 & C\theta & -S\theta C\phi \\ 0 & S\theta & C\theta C\phi \end{bmatrix} \quad (3.26)$$

et :

$${}^p\Omega_{RTL} = {}^p\mathbf{R}_b {}^b\Omega_{RTL} = \begin{bmatrix} C\phi C\psi & S\psi & 0 \\ -C\phi S\psi & C\psi & 0 \\ S\phi & 0 & 1 \end{bmatrix} \quad (3.27)$$

On en déduit alors l'expression du modèle cinématique opérationnel inverse :

$$\dot{\mathbf{q}}_a = \mathbf{J}_x^{-1} \dot{\Theta} \quad (3.28)$$

$$\mathbf{J}_x^{-1} = \mathbf{J}_p^{-1} \Omega_{RTL} \quad (3.29)$$

3.3.3 Calcul de la matrice jacobienne inverse

Dans la suite, on s'attachera à calculer la matrice jacobienne cinématique inverse \mathbf{J}_p^{-1} , le passage à la matrice jacobienne opérationnelle \mathbf{J}_x^{-1} étant obtenu en appliquant les résultats du paragraphe précédent.

Pour calculer \mathbf{J}_p^{-1} , on utilise la méthode proposée dans [Khalil et Ibrahim, 2007]. Comme pour le modèle géométrique inverse, l'idée de cette méthode est de transformer le module parallèle en trois robots séries connectés à la même plate-forme. Cette transformation est obtenue en coupant de manière virtuelle les liaisons aux points C_i et O ce qui a pour but d'isoler la plate-forme. Ainsi, les jambes 1 et 2 sont réduites à l'étude d'un robot série à 3ddl (articulation A_i et cardan B_i) alors que la jambe 3 est réduite à l'étude d'un robot série à 1ddl. Les lignes de la matrice jacobienne du module sont donc obtenues en calculant la vitesse de l'articulation motorisée à partir de la matrice jacobienne de chaque jambe.

Modèle cinématique inverse des jambes 1 et 2

Pour le calcul des deux premières lignes de la matrice jacobienne, on commence par exprimer la vitesse des points C_i en fonction de \mathbf{w}_p , soit :

$$\mathbf{V}_{C_i} = \mathbf{J}_{vi} \mathbf{w}_p \quad (3.30)$$

avec :

$$\mathbf{J}_{vi} = -\hat{\mathbf{P}}_{C_i}, i = 1, 2 \quad (3.31)$$

On peut également écrire le modèle cinématique d'une jambe :

$$\mathbf{V}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \quad (3.32)$$

où \mathbf{J}_i représente la matrice jacobienne (3×3) de la jambe i et \mathbf{V}_i la vitesse de son organe terminal.

Pour pouvoir obtenir le modèle cinématique complet de la jambe, on utilise la contrainte cinématique imposée par la rotule en C_i . La contrainte cinématique impose que seul le vecteur vitesse soit transmis par cette liaison (contrainte de position). On peut donc écrire que les équations (3.30) et (3.32) sont égales, soit :

$$\mathbf{J}_{vi} \mathbf{w}_p = \mathbf{J}_i \dot{\mathbf{q}}_i \quad (3.33)$$

En admettant que la matrice \mathbf{J}_i est inversible (l'étude des singularités est traitée dans [Chablat et Wenger, 2005]), la $i^{\text{ème}}$ ligne de la matrice jacobienne du module est donnée par :

$$\dot{\mathbf{q}}_i = \mathbf{J}_i^{-1}(1, :) (\mathbf{J}_{vi} \mathbf{w}_p) \quad (3.34)$$

avec $\mathbf{A}(i, :)$ représentant la i^{eme} ligne de la matrice \mathbf{A} .

Le calcul de \mathbf{J}_i^{-1} donne le résultat suivant pour $i=1,2$:

$${}^b\mathbf{J}_p^{-1}(i, :) = \begin{bmatrix} \frac{e_i N_i}{D_i} & -\frac{e_i C_{qi}}{D_i} & \frac{1}{D_i} \end{bmatrix} \hat{\mathbf{P}}_{Ci} \quad (3.35)$$

avec :

$$\begin{aligned} N_i &= \sqrt{2} + e_i S_{qi} \\ D_i &= C_{qi} (L + P_{Cix}) + e_i S_{qi} P_{Ciy} \\ \mathbf{P}_{Ci} &= \begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci} \\ {}^p\mathbf{P}_{Ci} &= \begin{bmatrix} 0 & 1 & -e_i \end{bmatrix}^T \end{aligned}$$

Modèle cinématique inverse de la jambe 3

Pour cette dernière jambe, l'équation de contrainte imposée par le cardan peut être écrite comme l'égalité de la vitesse angulaire de la plate-forme et de la jambe 3 lorsqu'elles sont projetées sur un axe \mathbf{h}_p qui est défini par la perpendiculaire commune aux deux axes du cardan \mathbf{h}_1 et \mathbf{h}_2 :

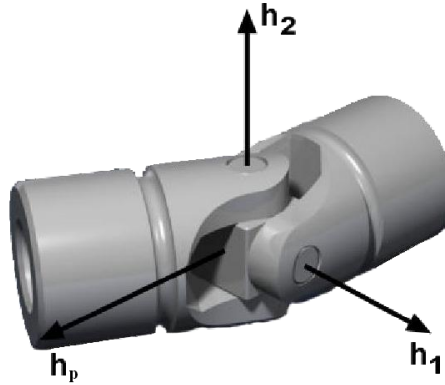


Figure 3.6: – Définition de l'axe de contrainte

L'équation résultant de cette contrainte s'écrit de la manière suivante dans le repère Σ_b :

$${}^b\mathbf{h}_p^T \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \dot{q}_3 = {}^b\mathbf{h}_p^T {}^b\mathbf{w}_p \quad (3.36)$$

ou dans le repère Σ_p :

$${}^p\mathbf{h}_p^T {}^p\mathbf{R}_b \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \dot{q}_3 = {}^p\mathbf{h}_p^T {}^p\mathbf{w}_p \quad (3.37)$$

On voit sur la figure (3.6) que \mathbf{h}_p s'obtient par le produit vectoriel des axes \mathbf{h}_1 et \mathbf{h}_2 . Dans le cas de notre module, les axes \mathbf{h}_1 et \mathbf{h}_2 sont définis respectivement par les colonnes 2 et 3 de la matrice $\mathbf{\Omega}_{RTL}$ (équations (3.27) ou (3.26) selon que l'on projette dans le repère Σ_p ou Σ_b). Ainsi, on peut écrire que :

$${}^b\mathbf{h}_p = \begin{bmatrix} C\phi & S\theta S\phi & -C\theta S\phi \end{bmatrix}^T \quad (3.38)$$

$${}^p\mathbf{h}_p = \begin{bmatrix} C\psi & -S\psi & 0 \end{bmatrix}^T \quad (3.39)$$

$$(3.40)$$

Dans la suite, on choisit d'exprimer le résultat dans le repère Σ_b . La troisième ligne de la matrice jacobienne du module k s'écrit donc :

$${}^b\mathbf{J}_p^{-1}(3, :) = \begin{bmatrix} 1 & S\theta \tan(\phi) & -C\theta \tan(\phi) \end{bmatrix} \quad (3.41)$$

La matrice jacobienne inverse complète s'écrit donc :

$${}^b\mathbf{J}_p^{-1} = \begin{bmatrix} \frac{-Cq_1 P_{C1z} - P_{C1y}}{D_1} & \frac{P_{C1x} - N_1 P_{C1z}}{D_1} & \frac{N_1 P_{C1y} + Cq_1 P_{C1x}}{D_1} \\ \frac{Cq_2 P_{C2z} - P_{C2y}}{D_2} & \frac{P_{C2x} + N_2 P_{C2z}}{D_2} & \frac{-N_2 P_{C2y} - Cq_2 P_{C2x}}{D_2} \\ 1 & S\theta \tan(\phi) & -C\theta \tan(\phi) \end{bmatrix} \quad (3.42)$$

avec : $\begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci}$.

Le calcul de la matrice jacobienne s'opère ensuite par inversion numérique de ${}^b\mathbf{J}_p^{-1}$.

3.4 Modèle cinématique inverse du second ordre

Le modèle cinématique inverse du second ordre permet de calculer les accélérations articulaires des actionneurs en fonction des vitesses \mathbf{w}_p et accélérations $\dot{\mathbf{w}}_p$ cartésiennes de la plate-forme du module. Pour calculer ce modèle, il suffit de dériver l'équation (3.23) ce qui donne :

$$\ddot{q}_i = \mathbf{J}_p^{-1}(i, :) \mathbf{w}_p + \dot{\mathbf{J}}_p^{-1}(i, :) \dot{\mathbf{w}}_p \quad (3.43)$$

Il est donc nécessaire ici de calculer $\dot{\mathbf{J}}_p^{-1}(i, :)$ pour $i=1, \dots, 3$.

Calcul de ${}^b\dot{\mathbf{J}}_p^{-1}(i, :)$ pour les jambes 1 et 2

A partir de l'équation (3.34), on peut en déduire la dérivée de la matrice jacobienne qui s'exprime par :

$${}^b\dot{\mathbf{J}}_p^{-1}(i, :) = \dot{\mathbf{J}}_i^{-1}(i, :) \mathbf{J}_{vi} + {}^b\mathbf{J}_i^{-1}(i, :) \dot{\mathbf{J}}_{vi} \quad (3.44)$$

A partir de l'équation (3.35), on peut en déduire l'expression de la dérivée de la matrice jacobienne :

$$\begin{aligned} {}^b\dot{\mathbf{J}}_p^{-1}(i, :) &= -\frac{\dot{D}_i}{D_i} {}^b\mathbf{J}_p^{-1}(i, :) + \frac{\dot{q}_i}{D_i} \begin{bmatrix} \mathbf{C}_{qi} & e_i \mathbf{S}_{qi} & 0 \end{bmatrix} \hat{\mathbf{P}}_{Ci} \\ &+ \begin{bmatrix} \frac{e_i N_i}{D_i} & -\frac{e_i C_{qi}}{D_i} & \frac{1}{D_i} \end{bmatrix} {}^b\hat{\mathbf{V}}_{Ci} \end{aligned} \quad (3.45)$$

avec :

$$\dot{D}_i = \dot{q}_i (-\mathbf{S}_{qi}(L + P_{Cix}) + e_i \mathbf{C}_{qi} P_{Ciy}) + \mathbf{C}_{qi} V_{Cix} + e_i \mathbf{S}_{qi} V_{Ciy} \quad (3.46)$$

$$\mathbf{P}_{Ci} = \begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci} \quad (3.47)$$

$${}^b\mathbf{V}_{Ci} = \begin{bmatrix} V_{Cix} & V_{Ciy} & V_{Ciz} \end{bmatrix}^T \quad (3.48)$$

Calcul de ${}^b\dot{\mathbf{J}}_p^{-1}(i, :)$ pour la jambe 3

La dérivée de la ième ligne de la matrice jacobienne dans le cas de la troisième jambe est immédiate. On obtient alors :

$${}^b\dot{\mathbf{J}}_p^{-1}(3, :) = \dot{\theta} \tan(\phi) \begin{bmatrix} 0 & \mathbf{C}\theta & \mathbf{S}\theta \end{bmatrix} + \dot{\phi} (1 + \tan(\phi)^2) \begin{bmatrix} 0 & \mathbf{S}\theta & -\mathbf{C}\theta \end{bmatrix} \quad (3.49)$$

3.5 Modélisation dynamique inverse du module

3.5.1 Définition du modèle dynamique pour les robots à structures fermées

Le modèle dynamique inverse (MDI) d'une structure fermée se calcule classiquement à partir du modèle dynamique d'une structure arborescente équivalente du robot. La structure arborescente équivalente d'un robot est obtenue à partir de la structure fermée par une ouverture virtuelle de chaque jambe (sauf une) au niveau d'une articulation passive (voir figure 3.7).

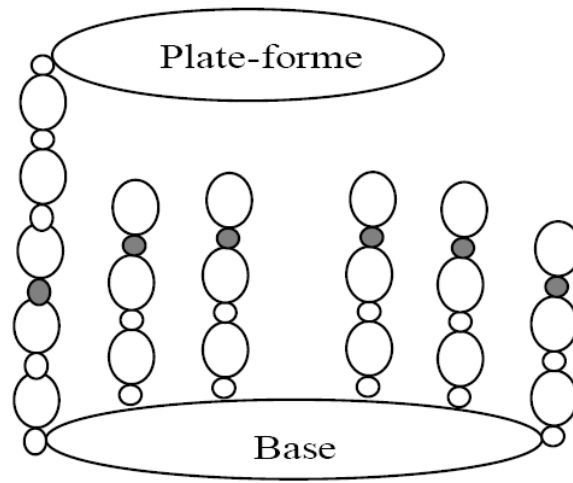


Figure 3.7: – Structure arborescente équivalente d'un robot parallèle

Le modèle dynamique s'écrit alors selon l'équation suivante [Kleinfinger et Khalil, 1986, Nakamura et Ghodoussi, 1988] :

$$\mathbf{\Gamma} = \mathbf{G}^T \mathbf{\Gamma}_{\text{ar}} \quad (3.50)$$

où :

- $\mathbf{\Gamma}_{\text{ar}}$ représente le modèle dynamique de la structure arborescente ouverte équivalente calculée en fonction des variables articulaires (\mathbf{q}_{ar} , $\dot{\mathbf{q}}_{\text{ar}}$, $\ddot{\mathbf{q}}_{\text{ar}}$) par l'algorithme de Newton-Euler ;
- \mathbf{G} représente la matrice jacobienne entre les variables de la structure arborescente \mathbf{q}_{ar} et les variables articulaires motorisées de la structure fermée \mathbf{q}_a et définie par :

$$\mathbf{G} = \frac{\partial \dot{\mathbf{q}}_{\text{ar}}}{\partial \dot{\mathbf{q}}_a} \quad (3.51)$$

Le problème de cette méthode est qu'il faut calculer le modèle dynamique de la plate-forme en fonction des articulations passives entre la plate-forme et l'organe terminal de la jambe auquel elle est connectée dans la structure arborescente. Pour éviter de compliquer

l'expression de ce modèle dynamique, une méthode proposée par [Ibrahim, 2006] consiste à séparer la plate-forme des différentes jambes et de calculer son modèle dynamique à part. On peut alors réécrire l'équation (3.50) sous une forme plus simple à calculer :

$$\mathbf{\Gamma} = \mathbf{J}_p^T \mathbb{F}_p + \sum_{i=1}^m \left(\frac{\partial \dot{\mathbf{q}}_{ar}}{\partial \dot{\mathbf{q}}_a} \right)^T \mathbf{H}_i \quad (3.52)$$

avec :

- \mathbb{F}_p : le torseur des efforts permettant de faire bouger la plate-forme et calculé par les équations de Newton-Euler ;
- \mathbf{J}_p : la matrice jacobienne du robot ;
- m : Le nombre de jambe(s) du robot ;
- \mathbf{H}_i : le modèle dynamique inverse de la jambe i .

3.5.2 Modèle dynamique inverse du module

Le module étudié dans le cadre de cette étude ne possède pas 6 degrés de liberté, mais seulement 3 degrés de liberté en rotation. L'équation (3.52) peut donc être simplifiée au cas d'un système à 3 ddl. On va donc chercher à exprimer le modèle dynamique inverse du module seulement en fonction des mouvements de rotation de la plate-forme : $\mathbf{\Gamma} = \mathbf{f}(\mathbf{}^b\mathbf{R}_p, \mathbf{w}_p, \dot{\mathbf{w}}_p)$. Sans considérer les forces extérieures au module, le nouveau modèle dynamique s'exprime par :

$$\mathbf{\Gamma} = \mathbf{J}_p^T \mathbf{M}_p + \sum_{i=1}^3 \left(\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} \right)^T \mathbf{H}_i \quad (3.53)$$

avec :

- $\mathbf{\Gamma}$: vecteur des couples articulaires ;
- \mathbf{M}_p : moment des forces extérieures appliquées à la plate-forme ;
- \mathbf{H}_i : le modèle dynamique inverse d'une jambe i du module. Il peut être calculé par une des méthodes dédiées aux robots séries [Luh *et al.*, 1980, Khalil et Kleinfinger, 1987] et en particulier la méthode récursive de Newton-Euler ;
- $\dot{\mathbf{q}}_i$: vecteur des vitesses articulaires de la jambe i du module ;
- $\dot{\mathbf{q}}_a$: vecteur vitesse des articulations motorisées du module défini par :

$$\dot{\mathbf{q}}_a = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix}^T \quad (3.54)$$

Le Moment \mathbf{M}_p des forces extérieures appliquées à la plate-forme s'obtient en appliquant les équations de Newton-Euler :

$$\mathbf{M}_p = \mathbf{I}_p \dot{\mathbf{w}}_p + \mathbf{w}_p \times (\mathbf{I}_p \mathbf{w}_p) \quad (3.55)$$

avec : \mathbf{I}_p la matrice d'inertie de la plate-forme.

Le calcul de $\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a}$ est obtenu en utilisant les propriétés de la structure parallèle :

$$\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} = \frac{\partial \dot{\mathbf{q}}_i}{\partial \mathbf{V}_i} \frac{\partial \mathbf{V}_i}{\partial \mathbf{w}_p} \frac{\partial \mathbf{w}_p}{\partial \dot{\mathbf{q}}_a} \quad (3.56)$$

avec : \mathbf{V}_i le vecteur vitesse cartésien de l'organe terminal de la jambe i .

L'équation (3.56) peut être réécrite sous forme d'une multiplication de matrices jacobiennes :

$$\frac{\partial \dot{\mathbf{q}}_i}{\partial \dot{\mathbf{q}}_a} = \mathbf{J}_i^{-1} \mathbf{J}_{vi} \mathbf{J}_p \quad (3.57)$$

avec :

- \mathbf{J}_i^{-1} la matrice jacobienne inverse de la chaîne cinématique i ,
- \mathbf{J}_{vi} la matrice jacobienne de projection qui permet d'exprimer les vitesses des organes terminaux des chaînes cinématiques \mathbf{V}_i en fonction de la vitesse de la plate-forme \mathbf{w}_p ,
- \mathbf{J}_p la matrice jacobienne cinématique du module.

L'équation (3.53) peut alors être réécrite dans une forme simplifiée :

$$\mathbf{\Gamma} = \mathbf{J}_p^T (\mathbf{M}_p + \mathbf{H}_b) \quad (3.58)$$

où \mathbf{H}_b représente la somme des modèles dynamiques des jambes exprimés dans l'espace cartésien de la plate-forme avec :

$$\mathbf{H}_b = \sum_{i=1}^3 \mathbf{J}_{vi}^T \mathbf{J}_i^{-T} \mathbf{H}_i \quad (3.59)$$

En tenant compte cette fois des inerties et frottements des actionneurs, le modèle dynamique du module est donné par :

$$\mathbf{\Gamma} = \mathbf{J}_p^T (\mathbf{M}_p + \mathbf{H}_b) + \mathbf{I}_a \ddot{\mathbf{q}}_a + \mathbf{F}_v \dot{\mathbf{q}}_a + \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}_a) \quad (3.60)$$

avec :

- \mathbf{I}_a la matrice diagonale (3×3) représentant les inerties des actionneurs ;
- \mathbf{F}_v la matrice diagonale (3×3) représentant les paramètres de frottements visqueux des actionneurs ;
- \mathbf{F}_s la matrice diagonale (3×3) représentant les paramètres de frottements secs des actionneurs.

Calcul du modèle dynamique des jambes H_b

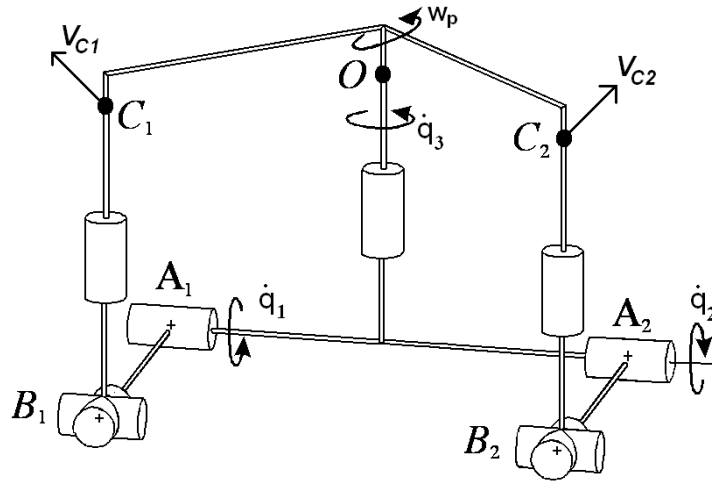


Figure 3.8: – Contraintes cinématiques du module

Pour pouvoir calculer H_b , il reste à exprimer les matrices J_i^{-1} et J_{vi} qui permettent de faire le lien entre la plate-forme et chacune des jambes. Les matrices J_i^{-1} peuvent être calculées en utilisant le logiciel SYMORO+ tandis que les matrices J_{vi} se déduisent à partir des contraintes cinématiques au niveau de l'organe terminal de chaque jambe (figure 3.8).

Pour les jambes ① et ② :

Pour les jambes ① et ②, on a vu que la contrainte cinématique consiste en une égalité (aux niveaux des points C_1 et C_2) entre la vitesse calculée par la plate-forme et la vitesse de l'organe terminal de la jambe (contrainte de position due à la rotule).

Or, l'expression de la vitesse des points C_i par la plate-forme s'exprime par :

$$\mathbf{V}_{Ci} = -\hat{\mathbf{P}}_{Ci} \mathbf{w}_p$$

La matrice jacobienne J_{vi} s'écrit donc :

$$\mathbf{J}_{vi} = -\hat{\mathbf{P}}_{Ci} \quad (3.61)$$

Pour la jambe ③ :

Pour cette jambe, le calcul de J_{vi} et J_i^{-1} est un peu plus compliqué puisque la contrainte cinématique imposée par le cardan est une contrainte bilatérale exprimée par rapport à l'axe \mathbf{h}_p (figure 3.6) (axe virtuel qui n'appartient ni à la jambe ni à la plate-forme) et

défini par l'équation (3.36) ou (3.37) selon qu'on exprime le résultat dans le repère Σ_b ou Σ_p .

En posant que la vitesse \mathbf{V}_3 de l'organe terminal de la jambe est la vitesse selon l'axe de contrainte \mathbf{h}_p , on obtient alors à partir de l'équation (3.36) :

$${}^b\mathbf{V}_3 = {}^b\mathbf{h}_p^T \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \dot{\mathbf{q}}_3 = {}^b\mathbf{h}_p {}^b\mathbf{w}_p \quad (3.62)$$

On peut en déduire immédiatement les expressions de \mathbf{J}_{v3} et J_3 :

$$\mathbf{J}_{v3} = {}^b\mathbf{h}_p \quad (3.63)$$

et

$$J_3 = {}^b\mathbf{h}_p^T \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \quad (3.64)$$

En remarquant que J_3 est un scalaire, l'équation (3.64) s'inverse immédiatement, permettant ainsi de calculer J_3^{-1} .

3.5.3 Simplification du modèle dynamique

Dans le cas de robots constitués de plusieurs jambes, le calcul du modèle dynamique peut s'avérer fastidieux du fait de l'obligation de calculer le modèle dynamique de chacune de ces jambes. Dans le cas particulier où l'on considère que les inerties des jambes du robot sont négligeables par rapport aux inerties de la plate-forme, on peut alors s'affranchir du calcul des modèles dynamiques des jambes et écrire que $\mathbf{H}_i = \mathbf{0}$. Par conséquent, dans un tel cas, le modèle dynamique inverse devient :

$$\mathbf{\Gamma} = \mathbf{J}_p^T \mathbf{M}_p + \mathbf{I}_a \ddot{\mathbf{q}}_a + \mathbf{F}_v \dot{\mathbf{q}}_a + \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}_a) \quad (3.65)$$

Dans le cadre de l'étude du robot anguille, les dimensions et masses des pièces du module parallèle tendent à faire penser qu'il est possible de négliger l'influence des jambes face aux inerties des deux pièces que constituent la base et la plate-forme. Le modèle CAO d'un module nous permet de connaître la masse de chaque corps et donc de faire un premier comparatif permettant de pré-valider notre hypothèse :

Ensemble de pièces	Masse de l'ensemble
Base du module	1,06 kg
Plate-forme sans électronique	324 g
Jambes ① et ②	20 g
Jambe ③	76 g

Tableau 3.2: – Masses des différents ensembles d'un module du prototype de l'anguille

Ce tableau montre bien la grande différence de poids entre les deux pièces principales et les pièces constituant chaque jambe. Pour connaître avec plus de précision l'influence des jambes sur le calcul des couples articulaires, on effectue la simulation du modèle dynamique inverse du module complet et on compare avec la simulation dans laquelle les modèles dynamiques des jambes sont négligés ($\mathbf{H}_i = \mathbf{0}$). Chaque modèle dynamique \mathbf{H}_i est calculé symboliquement en utilisant SYMORO+ [Khalil et Creusot, 1997].

Les paramètres inertiels calculés à partir du logiciel CATIA sont les suivants :

– Paramètres inertiels de la plate-forme par rapport à O :

$$\mathbf{MS} = \begin{bmatrix} 0,0237 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} 6,0105 \cdot 10^{-4} & 7,4857 \cdot 10^{-5} & 0 \\ 7,4857 \cdot 10^{-5} & 2,4 \cdot 10^{-3} & 0 \\ 0 & 0 & 2,2 \cdot 10^{-3} \end{bmatrix}$$

$$M = 0,324 \text{ kg}$$

– Paramètres inertiels des bielles $A_i B_i$ par rapport à A_i :

$$\mathbf{MS} = \begin{bmatrix} 2,8388 \cdot 10^{-4} \\ -1,3762 \cdot 10^{-5} \\ -1,3818 \cdot 10^{-5} \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} 1,3672 \cdot 10^{-6} & 3,3687 \cdot 10^{-7} & 1,0921 \cdot 10^{-6} \\ 3,3687 \cdot 10^{-7} & 8,9688 \cdot 10^{-6} & -1,018 \cdot 10^{-7} \\ 1,0921 \cdot 10^{-6} & -1,018 \cdot 10^{-7} & 7,7787 \cdot 10^{-6} \end{bmatrix}$$

$$M = 0,014 \text{ kg}$$

– Paramètres inertiels des biellettes $B_i C_i$ par rapport à B_i :

$$\mathbf{MS} = \begin{bmatrix} -1,2 \cdot 10^{-4} \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} 2,416 \cdot 10^{-8} & 0 & 0 \\ 0 & 4,211 \cdot 10^{-6} & 0 \\ 0 & 0 & 4,211 \cdot 10^{-6} \end{bmatrix}$$

$$M = 6,00 \cdot 10^{-3} \text{ kg}$$

- Paramètres inertiels de la jambe ③ :

$$\mathbf{J}(3,3) = \mathbf{ZZ}_3 = 4,58.10^{-6} \text{ M} = 0,076 \text{ kg}$$

A partir du modèle dynamique exprimé précédemment, nous avons réalisé une simulation sans tenir compte des paramètres dynamiques des 3 jambes, puis une autre avec tous les paramètres pour comparer. Le générateur de mouvements suivant est utilisé pour effectuer ces deux simulations :

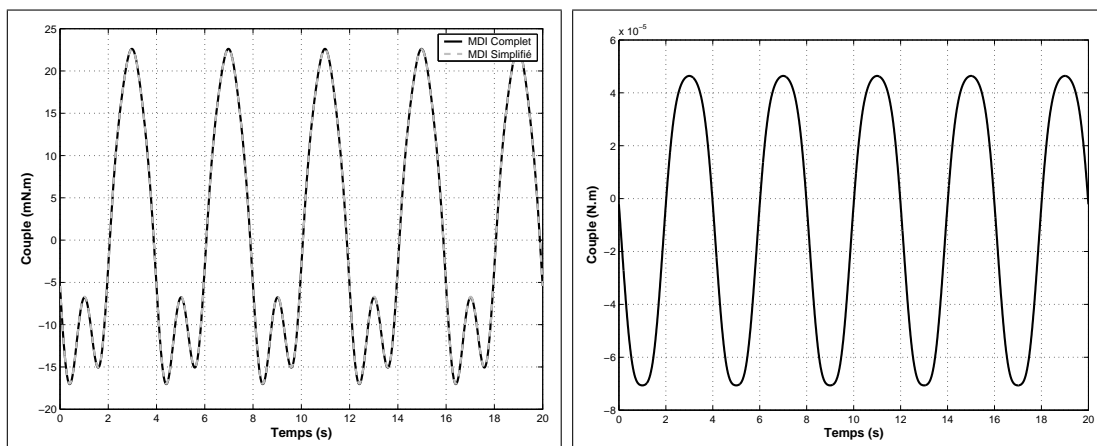
$$\theta(t) = \phi(t) = \psi(t) = A.\sin(2\pi\frac{t}{T_p}) \quad (3.66)$$

avec :

- $A = \pi/10$: l'amplitude du mouvement,
- $T_p = 4\text{s}$: la périodicité du mouvement.

Afin de ne prendre en compte que les effets dynamiques des jambes et voir s'ils ont une influence dans le calcul du modèle dynamique, on considère que la structure mécanique n'est pas soumise à la gravité.

Sous cette hypothèse, les figures (3.9 -a), (3.9 -c) et (3.9 -e) montrent les résultats de simulations pour le calcul des couples moteurs en utilisant le modèle dynamique complet et simplifié. Les figures (3.9 -b), (3.9 -d) et (3.9 -f) montrent la différence entre les modèles avec et sans prise en compte des paramètres dynamiques des jambes :



a - Couple Γ_1 pour le MDI simplifié et complet b - Différence de couple entre le MDI simplifié et complet

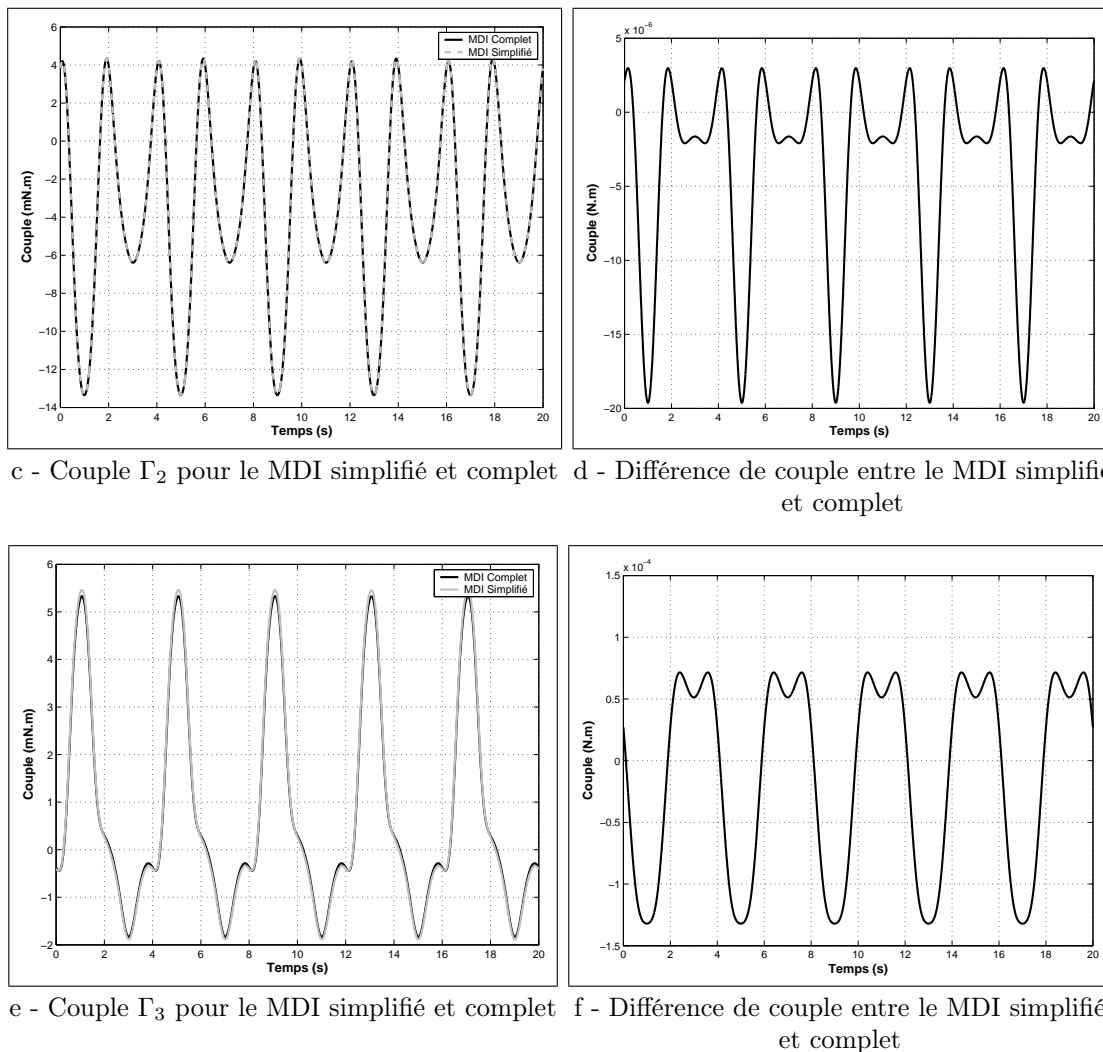


Figure 3.9: – Comparaison des couples pour les 3 articulations motorisées avec le Modèle Dynamique simplifié et Complet

Ces résultats montrent bien que la différence entre le modèle dynamique complet et simplifié est minime ce qui permet de valider l'hypothèse que l'on souhaite faire en négligeant l'influence des jambes du module. Il faut également considérer le fait que cette simulation est effectuée avec des mouvements très excitants, comparés à ceux qui seront utilisés pendant la nage de l'anguille, ce qui a pour conséquence d'augmenter les erreurs entre les deux modèles ainsi que la valeur des couples calculés. Si on réitère ces simulations pour des mouvements que fera l'anguille pendant la nage, on trouve alors une erreur maximale entre les deux modèles de l'ordre de 1%.

Durant cette étude, nous avons fait le choix de ne pas prendre en compte le terme de gravité dans les équations. Ce choix est justifié par le fait que le module étudié étant plongé dans l'eau, la poussée d'archimède viendra compenser le terme de gravité (sous hypothèse que le module est équilibré et qu'il flotte entre deux eaux).

3.6 Calcul de l'espace de travail

Le calcul de l'espace de travail permet de déterminer l'ensemble des configurations atteignables par la plate-forme du robot en tenant compte des limitations des actionneurs, des collisions ainsi que les limites de débattements des différentes pièces mécaniques. Pour le module étudié, on va analyser ces 3 types de limitations pour déterminer l'espace de travail.

Limitation des actionneurs :

Les moteurs utilisés dans ce robot sont des moteurs à courant continu et sans limite au niveau débattement angulaire.

Limites de débattements :

Parmi l'ensemble des pièces du module, deux types de pièces possèdent des limitations angulaires : le cardan situé en O et les rotules B_1 , B_2 , C_1 et C_2 (voir figure 3.10) :

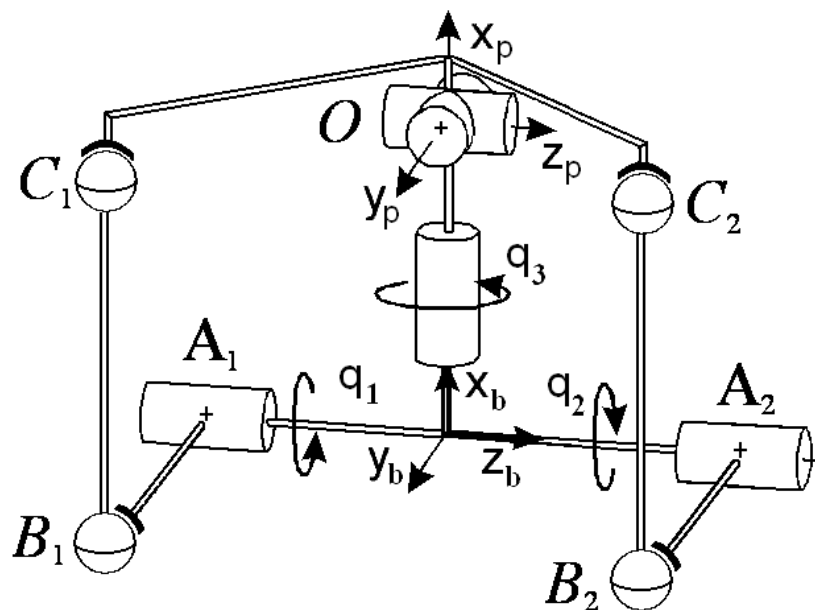


Figure 3.10: – Schéma cinématique équivalent à la mécanique du module

Usuellement, un cardan est utilisé pour un débattement de $\pm 45^\circ$ même s'il peut aller dans certaines configurations jusqu'à 90° . Pour les rotules, l'angle de débattement dépend des caractéristiques données par le constructeur. D'après la documentation, les rotules utilisées n'ont qu'un débattement angulaire de $\pm 22^\circ$.

Le respect de l'angle limite du cardan en O est simple à respecter puisqu'il est directement fonction des angles de Roulis-Tangage-Lacet (qui paramètrent l'orientation de la plate-

forme). Dans le cas des rotules, pour savoir si l'angle limite est atteint, il faut calculer le modèle géométrique inverse, calculer les coordonnées des points B_i et C_i et en déduire si les rotules respectent la contrainte de débattement angulaire maximum.

Dans notre cas, cette contrainte de débattement est caractérisée par un angle d'ouverture limite qui paramètre un cône duquel la biellette (B_iC_i) ne doit pas sortir. Le cône et l'angle limite sont représentés sur la figure (3.11 -a) :

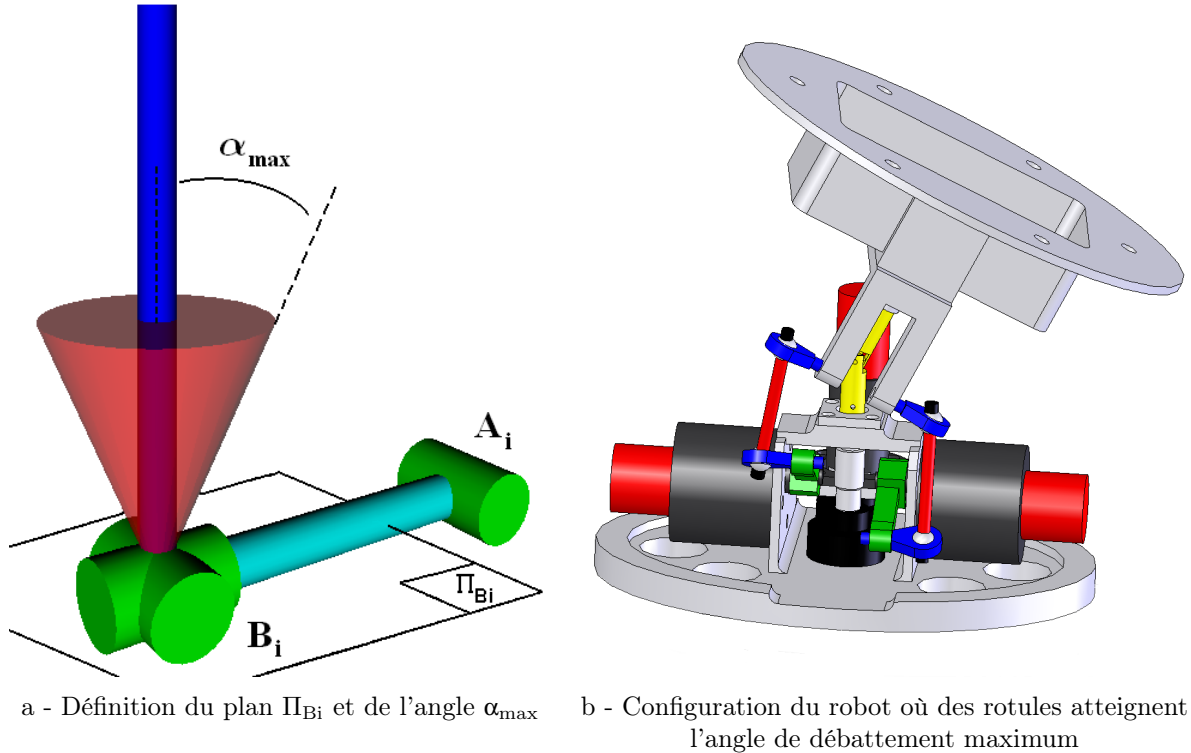


Figure 3.11: – Cône définissant le débattement des rotules B_i

A partir du modèle géométrique inverse décrit précédemment, on a pu calculer les coordonnées des points B_i et C_i . Pour déterminer l'angle d'ouverture des rotules, on définit les éléments géométriques suivants (pour $i=1,2$) :

- un plan Π_{B_i} défini par les axes $\overrightarrow{A_1A_2}$ et $\overrightarrow{A_iB_i}$;
- un plan Π_{C_i} défini par les axes $\overrightarrow{C_1C_2}$ et $\overrightarrow{OC_i}$;
- deux vecteurs unitaires \vec{n}_{B_i} et \vec{n}_{C_i} respectivement perpendiculaires aux plans Π_{B_i} et Π_{C_i} et d'origine B_i et C_i .

A présent, pour savoir si les biellettes (B_iC_i) sont dans les cônes, on calcule les quantités suivantes (pour $i=1,2$) :

$$- P_{S_{B_i}} = \langle \overrightarrow{B_iC_i}, \vec{n}_{B_i} \rangle / \|\overrightarrow{B_iC_i}\| ;$$

$$- \text{Ps}_{Ci} = \langle \overrightarrow{B_i C_i}, \overrightarrow{n_{Ci}} \rangle / \|\overrightarrow{B_i C_i}\|.$$

avec $\langle ., . \rangle$ désignant le produit scalaire entre deux vecteurs.

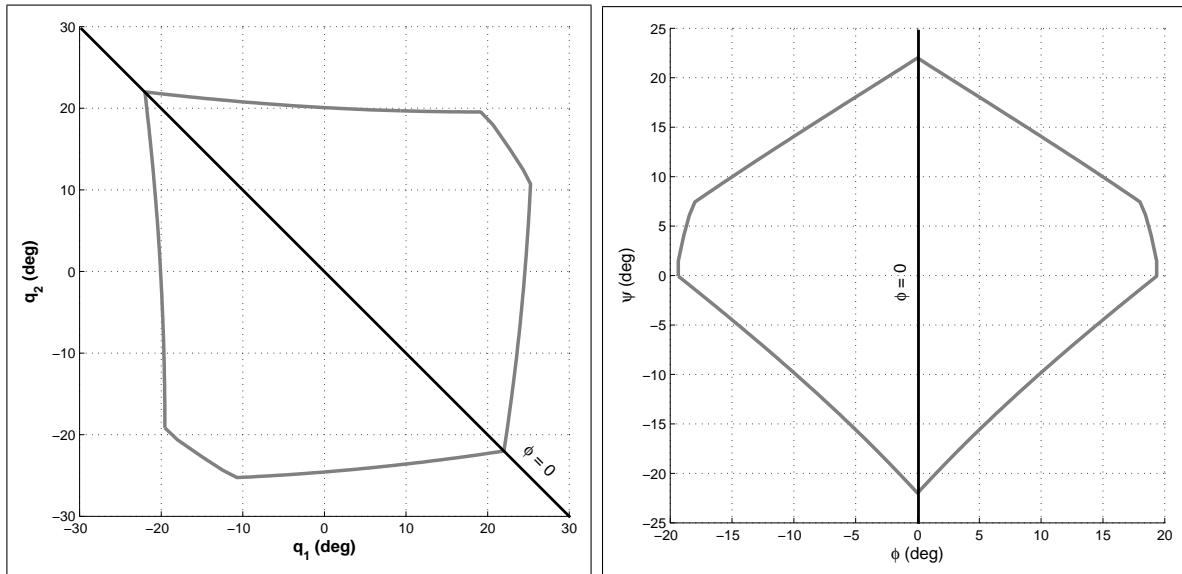
Si l'un des deux produits scalaires en valeur absolue est supérieur à $\cos(\alpha_{max})$ (avec α_{max} l'angle de débattement maximum) alors la biellette est en dehors d'un des cônes et par conséquent la configuration du module n'est pas atteignable.

Pour rappel, l'orientation de la plate-forme est paramétrée par les 3 angles de Roulis-Tangage-Lacet telle que :

- $\Theta = [\theta \ \phi \ \psi]^T$ est le vecteur des 3 angles de cardan ;
- la matrice d'orientation de la plate-forme par rapport à la base est définie par :

$${}^b\mathbf{R}_p = \mathbf{Rot}(\mathbf{x}, \theta) \mathbf{Rot}(\mathbf{y}, \phi) \mathbf{Rot}(\mathbf{z}, \psi)$$

Les figures (3.12-a) et (3.12-b) montrent l'espace de travail articulaire et cartésien obtenu pour un angle de roulis nul soit $\theta = 0$:



a - Espace de travail articulaire

b - Espace de travail cartésien

Figure 3.12: – Espace de travail d'un module pour $\theta = 0$

L'espace de travail que l'on obtient après calcul des configurations atteignables possède plusieurs propriétés. Tout d'abord, on peut voir qu'il existe un axe de symétrie paramétré par $\phi = 0$ (angle de lacet nul) et qui correspond physiquement au plan de symétrie (plan $OX_b Y_b$) du module de l'anguille. Deuxièmement, on peut voir sur la figure (3.12-b) que

les limites de l'espace de travail cartésien sont de $\pm 22^\circ$, ce qui correspond à la limite d'angle des rotules. Il est également important de noter que l'espace de travail ne présente pas de discontinuités et que par conséquent le passage d'une configuration à une autre ne posera pas de problème majeur.

Pour étudier l'influence du roulis (angle θ) sur l'espace de travail, on montre sur les figures (3.13 -a) et (3.13 -b) l'espace de travail obtenu pour deux valeurs symétriques de cet angle :

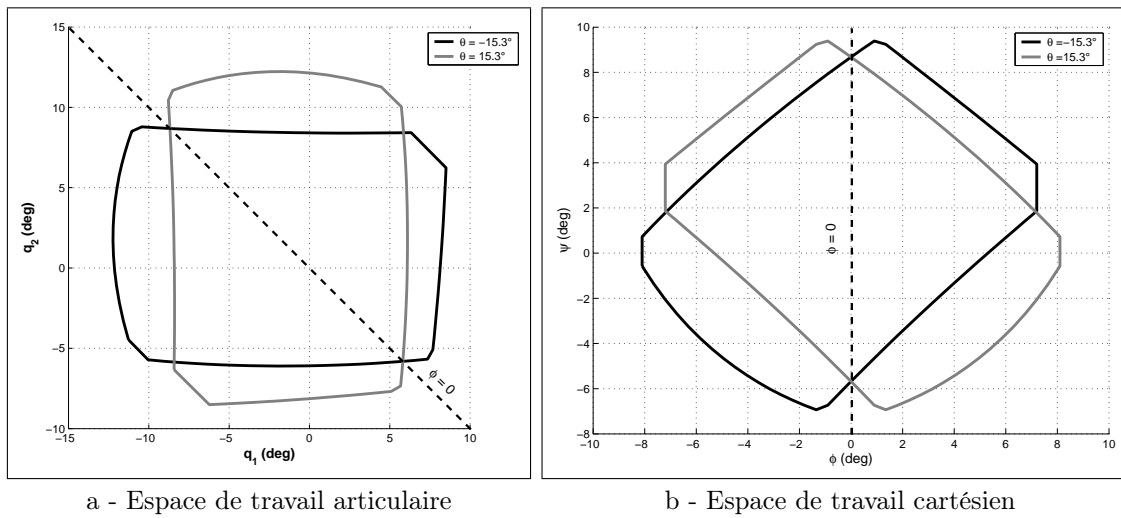


Figure 3.13: – Espace de travail d'un module pour $\theta = \pm 15.3^\circ$

On retrouve cette fois encore un axe de symétrie $\phi = 0$ pour les deux espaces de travail. On remarque également que la propriété de continuité de l'espace est conservée mais que les bornes de cet espace ont diminué (environ 9° comparés aux 22° qu'on avait précédemment). Finalement, en étudiant l'espace de travail complet, on observe qu'il possède un axe de symétrie paramétré par $(\phi = 0, \theta = 0)$ dans l'espace cartésien et $(q_2 = -q_1, q_3 = 0)$ dans l'espace articulaire.

Les figures (3.14 -a) et (3.14 -b) représentent l'espace de travail du module étudié et représentés respectivement dans l'espace articulaire et l'espace cartésien :

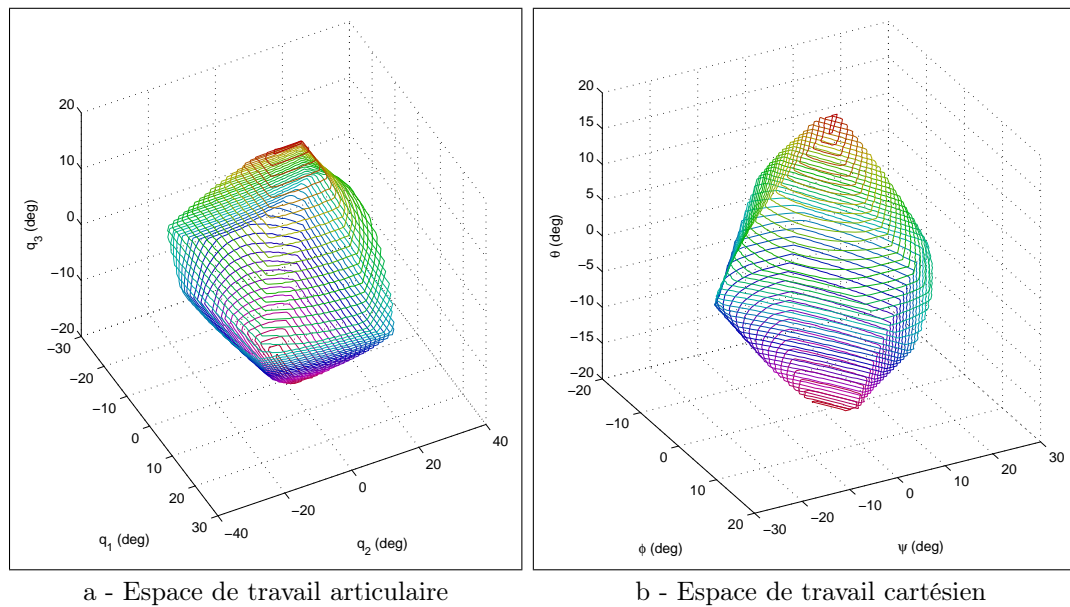


Figure 3.14: – Espace de travail complet du module

Limites de collision :

Les collisions entre pièces provoquent une autre limitation à prendre en compte dans l'étude de l'espace de travail. En effet, ces limites peuvent provoquer la détérioration ou la destruction de pièces. Dans le cas présent, les collisions pouvant se produire se situent entre la plate-forme et la base, ou entre l'ensemble (bielle-biellette) et la base. L'ensemble mécanique formant ce qu'on appelle la base est représenté sur la figure (3.15) :

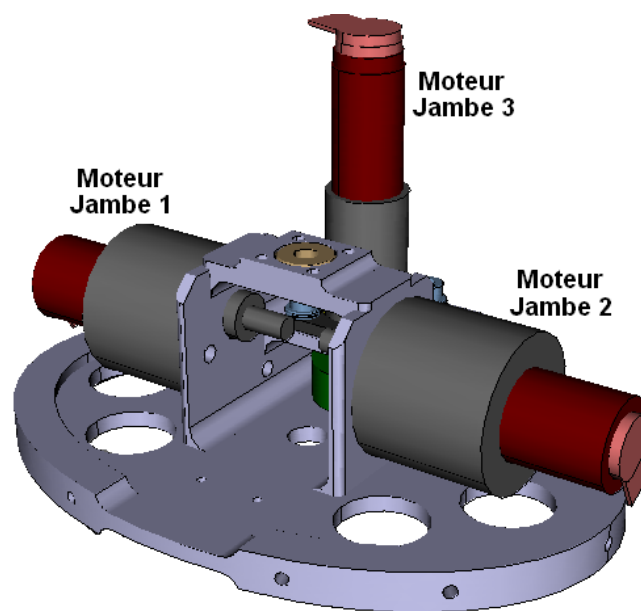


Figure 3.15: – Description de la base du module

Cette base est constituée d'un bloc dont le plan supérieur est horizontal et d'une chape sur laquelle sont fixés les 3 moteurs permettant de faire bouger la plate-forme. Sur la figure (3.15), on remarque tout de suite que le moteur de la jambe 3 (également appelé moteur de Roulis) étant fixé en vertical et de dimension assez importante, il risque de provoquer une collision avec la plate-forme.

Si on ne tient pas compte de la plate-forme pour le moment, deux types de collisions peuvent avoir lieu entre l'ensemble bielle-biellette et la base :

- une collision entre les rotules B_i et le plan horizontal de la base ;
- une collision entre les rotules B_i et la chape supportant les moteurs.

Les figures (3.16 -a) et (3.16 -b) permettent de mieux appréhender les deux situations :

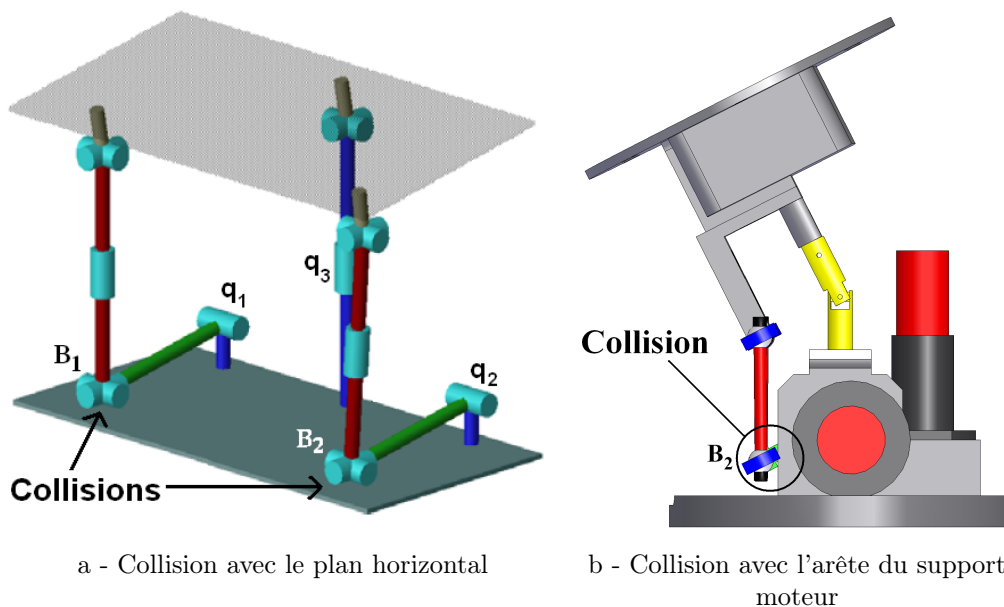


Figure 3.16: – Intersections de l'ensemble bielle-biellette avec la base

On peut remarquer sur la figure (3.16 -b) que les rotules B_i entrent en collision avec la chape supportant les moteurs avant de rentrer en collision avec le plan horizontal de la base. Par conséquent, la configuration de la figure (3.16 -a) ne peut se produire.

Par le biais de la modélisation sous le logiciel Catia, on détermine que ce type de configuration se produit lorsque l'angle des articulations q_1 ou q_2 est de $\pm 40.4^\circ$. Or, compte tenu de l'étude précédente, ce cas de figure ne peut pas être atteint car l'angle des articulations se situe en dehors de l'espace de travail que l'on a trouvé. L'étude de ces collisions montre donc que l'espace de travail n'est pas réduit.

Il reste maintenant à étudier le(s) cas de collision(s) entre la plate-forme et les éléments de la base. En réalité, le seul cas de collision qui peut se produire est celui où la plate-forme entre en contact avec le corps du moteur de la jambe 3 (moteur de roulis) comme présenté sur la figure (3.17) :

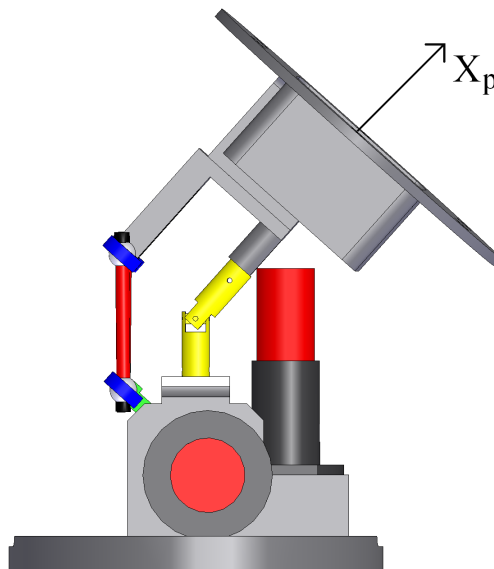


Figure 3.17: – Intersection entre la plate-forme et le moteur de roulis

Le moteur de roulis n'étant pas positionné de manière fixe (il est monté sur un excentrique qui permet de régler l'accostage entre deux roues dentées), on va donc se placer dans le cas le plus défavorable pour étudier la collision. Ce cas est obtenu lorsque le moteur de roulis se situe au plus près de l'axe central du module.

Ce problème de collision entre l'axe fixé au cardan (fixé également à la plate-forme) et le corps du moteur peut être ramené à l'étude de l'intersection entre une droite (l'axe x_p) et un cylindre (le corps du moteur). Les résultats de cette étude montrent que l'espace de travail obtenu précédemment n'est pas réduit.

La conclusion de cette étude montre que seule la condition de l'angle limite d'ouverture des rotules B_i et C_i est à prendre en compte dans l'étude de l'espace de travail du module. Cette constatation est très intéressante car le calcul de cet angle demande peu d'opérations mathématiques en utilisant le modèle géométrique inverse. Le calcul de ce critère peut être également intégré dans l'informatique embarquée en tant que butée logiciel et ainsi assurer que les collisions ne puissent se produire quelles que soient les consignes appliquées pour les angles de Roulis-Tangages-Lacet.

3.7 Conclusion

Nous avons étudié dans ce chapitre une structure mécanique parallèle à 3 degrés de liberté conçue dans le cadre du projet robot anguille. A travers cette étude, nous avons développé les modèles géométrique, cinématique et dynamique de ce module qui seront utilisés pour la modélisation du prototype complet au chapitre 4. Ainsi, nous avons pu déterminer l'espace de travail et montrer que celui-ci était admissible vis-à-vis de la plage de fonctionnement prévue. Nous avons également montré que le modèle dynamique d'une structure parallèle peut être simplifié du fait que les inerties des jambes sont négligeables comparées à celles de la plate-forme. Ce dernier résultat est particulièrement intéressant puisqu'il permet de simplifier le modèle dynamique de la structure complète du robot.

Finalement, l'étude de ce module permet de montrer deux aspects fondamentaux de la conception mécanique d'un robot. Premièrement, on a pu voir que le choix des proportions dans la géométrie du module permettait de simplifier le calcul du modèle géométrique. Mais dans un second temps, on s'est aperçu en étudiant l'espace de travail que le choix des dimensions réelles du module est important car il peut influencer son comportement. Dans le cas présent, les dimensions ont été étudiées de telle manière que les collisions entre pièces soient les moins nombreuses possible et ne modifient pas l'espace de travail.

Chapitre 4

Modèle Dynamique du prototype hybride

4.1 Introduction

Dans ce chapitre est présenté le modèle dynamique du prototype de robot anguille réalisé au sein du laboratoire de l'IRCCyN. Il s'agit ici d'un robot que l'on classe dans la catégorie des robots hybrides, c'est-à-dire composé d'un ensemble de structures parallèles connectées à la chaîne de manière série. Le robot anguille étudié se compose de 12 structures parallèles (appelées par la suite modules) à 3 ddl de type rotule que nous avons présenté dans le chapitre 3. Les 12 modules sont identiques en structure et en dimensions et sont chacun actionnés par 3 moteurs à courant continu. Ce type de robot hybride est particulièrement intéressant puisqu'il combine les avantages des structures séries pour leur large espace de travail et les avantages des structures parallèles pour leur robustesse et leur précision :

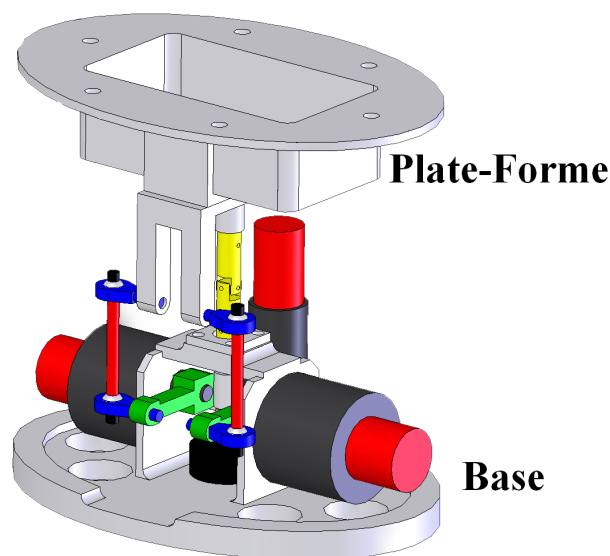


Figure 4.1: – Model CAO d'un module seul

Afin d'optimiser la place disponible pour les différents circuits électroniques, chaque module est connecté en opposition par rapport à son précédent. Ainsi, le robot est constitué d'une succession de blocs, tantôt comprenant les modules d'informatique et d'électronique embarqués, tantôt comprenant l'ensemble des moteurs permettant d'actionner deux modules successifs. L'agencement des modules sur l'anguille est représenté sur la figure (4.2) :

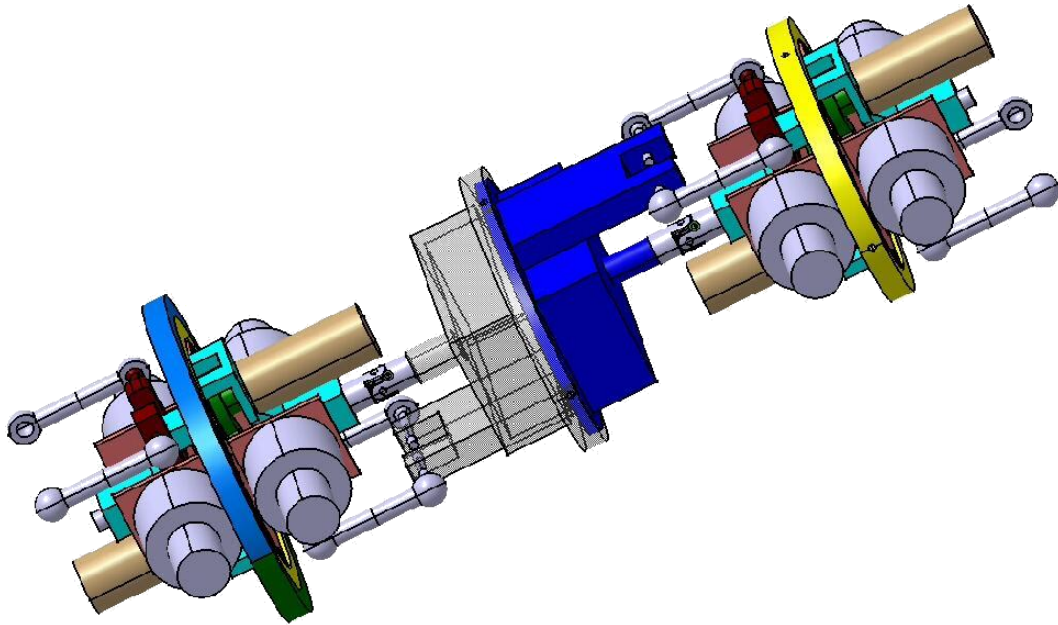


Figure 4.2: – Positionnement des modules dans le prototype

En faisant le lien avec la modélisation de chaque module, on remarquera que les plateformes et les bases sont respectivement connectées ensemble sauf pour le module 1 dont la base est connectée à la tête et le module 12 dont la plateforme est connecté à la queue du robot. La structure du robot ainsi définie est présentée sur la figure (4.3) :

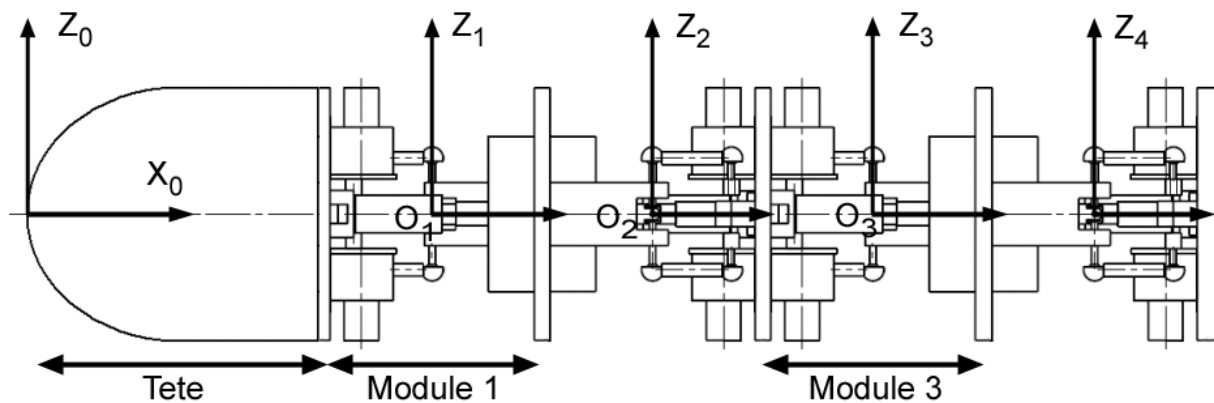


Figure 4.3: – Assemblage des modules

La tête de l'anguille, qui est la base du premier module, est numérotée comme le corps 0. Les modules sont ensuite numérotés de 1 à n , puis la queue qui est la plate-forme du module n est numérotée $n+1$. On assigne un repère Σ_k d'origine O_k lié à la plate-forme de chaque module k , de telle manière que l'axe \mathbf{x}_k soit pris tangent à la fibre neutre de l'anguille et l'axe \mathbf{z}_k soit selon l'axe de lacet, comme défini sur la figure (4.3). Pour chaque module k , on fixe également un repère Σ_{k-1} de centre O_{k-1} lié à sa base dont les axes sont définis de la même manière. La distance entre les origines O_{k-1} et O_k est notée d_k et représente une translation pure selon l'axe \mathbf{x}_{k-1} . On remarquera que le repère Σ_{k-1} lié à la base du module k représente également le repère lié à la plate-forme du module $k-1$ (figure 4.4)

Pour la modélisation dynamique de ce robot, on utilise les algorithmes récursifs de Newton-Euler pour leur simplicité d'utilisation et de programmation. Dans le cas du modèle dynamique inverse, l'algorithme développé ici permet de calculer l'accélération de la base du robot (ici la tête de l'anguille) ainsi que les couples articulaires en fonction des positions, vitesses et accélérations cartésiennes des plates-formes des modules parallèles et de la vitesse de la tête [Gallot *et al.*, 2007]. Pour le modèle dynamique direct, l'algorithme permet de calculer l'accélération de la tête de l'anguille et les accélérations cartésiennes des plates-formes en fonction des couples articulaires et des positions et vitesses cartésiennes des plates-formes des modules. Ces algorithmes constituent respectivement la généralisation des algorithmes de Luh [Luh *et al.*, 1980] et de Featherstone [Featherstone, 1983] des robots séries à base fixe.

Pour plus de facilité dans la commande de l'anguille, les consignes de positions, vitesses et accélérations se feront dans l'espace opérationnel plutôt que dans l'espace articulaire. Ce choix a également pour avantage de pouvoir exprimer les torseurs cinématiques sans avoir recours à la matrice jacobienne d'un module puisqu'ils peuvent directement être exprimés à partir des angles de Roulis-Tangage-Lacet et de leurs dérivées successives.

Ce chapitre est organisé de la manière suivante : l'anguille étant composée de modules identiques mais dont certains sont ce que l'on appelle "à l'endroit" et d'autres "à l'envers", on présentera dans les paragraphes 4.2 et 4.3 les relations qu'il y a entre les modèles géométriques et cinématiques pour ces deux types de modules. Nous présentons ensuite dans les paragraphes 4.4 et 4.5 les modèles dynamiques inverse et direct du prototype de l'anguille. Enfin, dans les paragraphes 4.6 et 4.7 nous présentons des résultats de simulations de nos algorithmes et des comparaisons avec les résultats obtenus au chapitre 2, puis nous présentons la modélisation du robot lorsque celui-ci est équipé d'ailerons.

4.2 Modélisation géométrique du prototype

L'anguille est donc composée de modules identiques (étudiés dans la chapitre 3) mais assemblés tête-bêche les uns par rapport aux autres. Les modules n'étant pas agencés de la même manière, il va falloir en étudier l'impact sur les modèles géométriques afin que ceux-ci soient compatibles avec les nouveaux repères Σ_k du robot. Les deux configurations possibles (à l'endroit et à l'envers) des modules sont représentées sur la figure (4.4) :

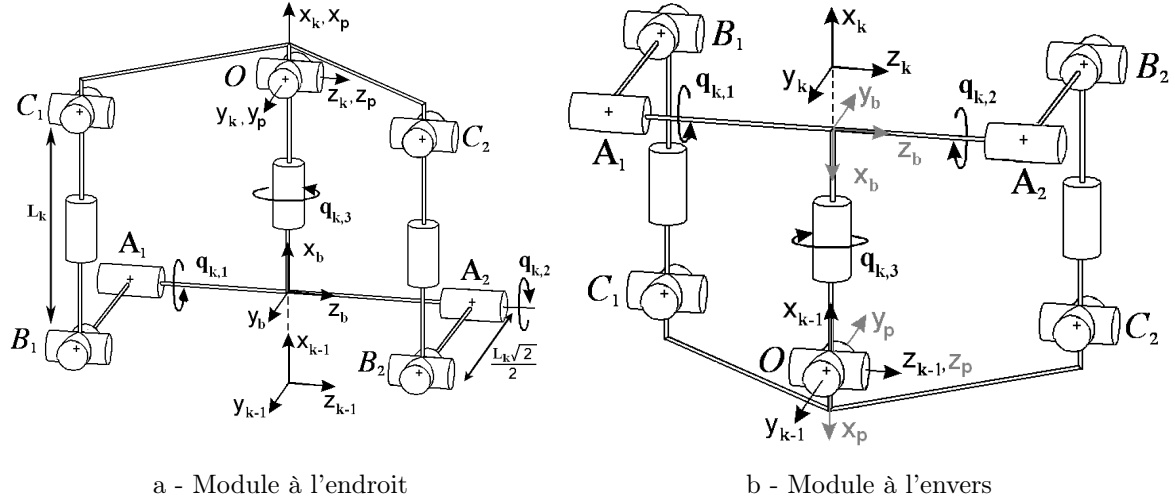


Figure 4.4: – Placement des repères Σ_k et Σ_{k-1} des modules à l'endroit et à l'envers de l'anguille

On peut tout de suite remarquer que pour le module à l'endroit (figure 4.4 -a) le repère Σ_p défini dans le chapitre 3 correspond au nouveau repère Σ_k et que le repère Σ_{k-1} est obtenu par une translation du repère Σ_b . Par conséquent, les résultats obtenus durant l'étude précédente pourront être utilisés tels quels. Cependant, pour le module à l'envers, les repères sont complètement différents ce qui nous oblige à adapter les modèles géométrique et cinématique du chapitre précédent.

Modèle géométrique du module à l'envers

On rappelle que pour tous les modules, l'orientation de la plate-forme est exprimée à partir du vecteur :

$$\Theta_k = [\theta_k \quad \phi_k \quad \psi_k]^T \quad (4.1)$$

On peut alors définir la matrice d'orientation de la plate-forme d'un module k par rapport à sa base :

$${}^{k-1}\mathbf{R}_k = \begin{bmatrix} C\phi_k C\psi_k & -C\phi_k S\psi_k & S\phi_k \\ S\theta_k S\phi_k C\psi_k + C\theta_k S\psi_k & -S\theta_k S\phi_k S\psi_k + C\theta_k C\psi_k & -S\theta_k C\phi_k \\ -C\theta_k S\phi_k C\psi_k + S\theta_k S\psi_k & C\theta_k S\phi_k S\psi_k + S\theta_k C\psi_k & C\theta_k C\phi_k \end{bmatrix} \quad (4.2)$$

A partir de l'étude d'un module seul présenté dans le chapitre 3, nous avons pu établir la relation entre la situation de la plate-forme via la matrice de rotation et les variables articulaires motorisées du module :

$$\mathbf{q}_k = \mathbf{f} \left({}^b\mathbf{R}_p \right) = \mathbf{f} \left({}^{k-1}\mathbf{R}_k \right) \quad (4.3)$$

Pour rappel, dans le cas du module à l'endroit, on peut directement remplacer la matrice d'orientation ${}^b\mathbf{R}_p$ par la matrice ${}^{k-1}\mathbf{R}_k$.

Dans le cas du module dit "à l'envers", le module a en réalité subi une rotation d'un angle π autour de l'axe \mathbf{z}_k . Dans ce cas de figure, on remarque que la plate-forme du module est devenue la base et vice-versa. Pour calculer le modèle géométrique inverse de ce module, une solution consiste donc à réexprimer la matrice de rotation ${}^b\mathbf{R}_p$ du module seul, en fonction des différents changements de repères et de la matrice de rotation ${}^{k-1}\mathbf{R}_k$.

Les matrices d'orientation qui permettent de passer respectivement des repères Σ_k et Σ_{k-1} aux repères Σ_b et Σ_p s'expriment de la manière suivante (selon les configurations montrées sur la figure 4.4 -b) :

$${}^b\mathbf{R}_k = {}^p\mathbf{R}_{k-1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

On peut donc exprimer la matrice ${}^b\mathbf{R}_p$ du module "à l'envers" en fonction de la matrice ${}^k\mathbf{R}_{k-1}$ qui sera utilisée dans l'algorithme de Newton-Euler :

$${}^b\mathbf{R}_p = {}^b\mathbf{R}_k {}^{k-1}\mathbf{R}_k^T {}^{k-1}\mathbf{R}_p \quad (4.5)$$

soit :

$${}^b\mathbf{R}_p = \begin{bmatrix} A_{11} & A_{21} & -A_{31} \\ A_{12} & A_{22} & -A_{32} \\ -A_{13} & -A_{23} & A_{33} \end{bmatrix}$$

avec A_{ij} la composante située sur la i^{eme} ligne et j^{eme} colonne de la matrice ${}^{k-1}\mathbf{R}_k$.

En reprenant les résultats du chapitre 3, on en déduit donc que le modèle géométrique inverse d'un module s'écrit de la manière suivante :

- **Pour les articulations $q_{k,1}$ et $q_{k,2}$:**

On cherche tout d'abord à trouver la solution de l'équation du second degré suivante :

$$\alpha_{i1} Q_i^2 + \alpha_{i2} Q_i + \alpha_{i3} = 0 \quad (4.6)$$

avec :

$$\alpha_{i1} = 2 + P_{Ciy} + e_i P_{Ciz} + \sqrt{2} P_{Cix} \quad (4.7)$$

$$\alpha_{i2} = 2e_i \left(\sqrt{2} + P_{Cix} \right) \quad (4.8)$$

$$\alpha_{i3} = 2 - P_{Ciy} + e_i P_{Ciz} + \sqrt{2} P_{Cix} \quad (4.9)$$

$$e_1 = 1 \quad \text{et} \quad e_2 = -1 \quad (4.10)$$

et :

- $\begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^{k-1}\mathbf{R}_k {}^p\mathbf{P}_{Ci}$, pour un module à l'endroit ;
- $\begin{bmatrix} P_{Cix} & P_{Ciy} & P_{Ciz} \end{bmatrix}^T = {}^b\mathbf{R}_k {}^{k-1}\mathbf{R}_k^T {}^{k-1}\mathbf{R}_p {}^p\mathbf{P}_{Ci}$, pour un module à l'envers ;
- ${}^p\mathbf{P}_{Ci} = \begin{bmatrix} 0 & 1 & -e_i \end{bmatrix}^T$.

Ensuite, les articulations $q_{k,1}$ et $q_{k,2}$ sont obtenues par les relations suivantes :

$$q_{k,i} = 2 \left(\text{atan}(Q_i) \right) \quad (4.11)$$

$$Q_i = \frac{-\alpha_{i2} + e_i \sqrt{\alpha_{i2}^2 - 4\alpha_{i1}\alpha_{i3}}}{2\alpha_{i1}} \quad (4.12)$$

- **Pour l'articulation $q_{k,3}$:**

Pour un module, à l'endroit, on a vu que $q_{k,3} = \theta_k$. Il est également possible d'exprimer cette relation en fonction des composantes de la matrice de rotation ${}^{k-1}\mathbf{R}_k$, soit :

$$\theta_k = \text{atan2}(-{}^b\mathbf{R}_p(2,3), {}^b\mathbf{R}_p(3,3)) \quad (4.13)$$

avec ${}^b\mathbf{R}_p = {}^{k-1}\mathbf{R}_k$.

Pour le modèle à l'envers, ayant exprimé la nouvelle matrice de rotation entre sa plateforme et sa base, on en déduit que :

$$q_{k,3} = \text{atan2}(-{}^b\mathbf{R}_p(2,3), {}^b\mathbf{R}_p(3,3)) \quad (4.14)$$

avec ${}^b\mathbf{R}_p = {}^b\mathbf{R}_k {}^{k-1}\mathbf{R}_k^T {}^{k-1}\mathbf{R}_p$.

4.3 Modèles cinématiques des modules

Pour chaque module de l'anguille, le modèle cinématique inverse permet de calculer la vitesse des articulations motorisées en fonction de la vitesse de sa plate-forme. Comme pour le modèle géométrique, le cas du module à l'endroit correspond à l'étude faite au chapitre 3 : on repartira donc de ces résultats afin d'exprimer les modèles cinématiques des modules à l'envers.

4.3.1 Modèle cinématique inverse

Pour le modèle cinématique du module à l'endroit, on a vu dans le paragraphe (3.3.3) qu'il pouvait s'écrire sous la forme :

$$\dot{\mathbf{q}}_{ak} = {}^b\mathbf{J}_p^{-1} {}^b\mathbf{w}_{p/b} \quad (4.15)$$

avec :

$${}^b\mathbf{J}_p^{-1} = \begin{bmatrix} \frac{-Cq_1P_{C1z} - P_{C1y}}{D_1} & \frac{P_{C1x} - N_1P_{C1z}}{D_1} & \frac{N_1P_{C1y} + Cq_1P_{C1x}}{D_1} \\ \frac{Cq_2P_{C2z} - P_{C2y}}{D_2} & \frac{P_{C2x} + N_2P_{C2z}}{D_2} & \frac{-N_2P_{C2y} - Cq_2P_{C2x}}{D_2} \\ 1 & S\theta \tan(\phi) & C\theta \tan(\phi) \end{bmatrix} \quad (4.16)$$

et :

$${}^b\mathbf{J}_p^{-1} = \mathbf{f}({}^b\mathbf{R}_p, \theta, \phi), \quad \dot{\mathbf{q}}_{ak} = \begin{bmatrix} \dot{q}_{k,1} & \dot{q}_{k,2} & \dot{q}_{k,2} \end{bmatrix}$$

En rappelant que les repères Σ_k et Σ_p sont confondus et que le repère Σ_{k-1} est obtenu par une translation pure du repère Σ_b , on en déduit que :

$${}^b\mathbf{J}_p^{-1} = {}^{k-1}\mathbf{J}_k^{-1}, \quad {}^b\mathbf{w}_{p/b} = {}^{k-1}\mathbf{w}_k, \quad {}^b\mathbf{R}_p = {}^{k-1}\mathbf{R}_k, \quad \theta = \theta_k, \quad \phi = \phi_k$$

Dans le cas où on préfère exprimer la matrice jacobienne \mathbf{J}_k^{-1} dans le repère Σ_k plutôt que dans le repère Σ_{k-1} , on en déduit que :

$${}^k\mathbf{J}_k^{-1} = {}^{k-1}\mathbf{J}_k^{-1} {}^{k-1}\mathbf{R}_k \quad (4.17)$$

Application au cas du module à l'envers

Dans le cas du module à l'envers, on a vu que les repères Σ_{k-1} et Σ_k ne correspondent plus avec les repères Σ_b et Σ_p . Par conséquent, on est obligé de recalculer la matrice de rotation ${}^b\mathbf{R}_p$ et les angles θ et ϕ pour qu'ils correspondent à la nouvelle situation du module. On aura donc ${}^b\mathbf{J}_p^{-1} = \mathbf{f}({}^b\mathbf{R}_p, \theta, \phi)$ avec :

- la nouvelle matrice de rotation ${}^b\mathbf{R}_p$ est obtenue en utilisant l'équation (4.5) ;
- l'angle θ est obtenu en utilisant la relation (4.13) ;
- l'angle ϕ se déduit de la même manière en utilisant l'équation suivante :

$$\phi_k = \text{atan2}({}^b\mathbf{R}_p(1, 3), C\theta_k {}^b\mathbf{R}_p(3, 3) - S\theta_k {}^b\mathbf{R}_p(2, 3)) \quad (4.18)$$

Le modèle cinématique inverse du module à l'envers s'écrit donc de la même manière :

$$\dot{\mathbf{q}}_{ak} = {}^b\mathbf{J}_p^{-1} {}^b\mathbf{w}_{p/b} = - {}^b\mathbf{J}_p^{-1} {}^b\mathbf{w}_{b/p} \quad (4.19)$$

Pour compléter cette équation, il nous faut encore exprimer la vitesse ${}^b\mathbf{w}_{b/p} = \mathbf{f}({}^{k-1}\mathbf{w}_k)$ en tenant compte des changements de repères soit :

$${}^{k-1}\mathbf{w}_k = {}^{k-1}\mathbf{R}_p {}^p\mathbf{R}_b {}^b\mathbf{w}_{b/p} \quad (4.20)$$

En substituant (4.5) dans cette équation, on obtient alors :

$${}^{k-1}\mathbf{w}_k = {}^{k-1}\mathbf{R}_p ({}^{k-1}\mathbf{R}_p^T {}^{k-1}\mathbf{R}_k {}^b\mathbf{R}_k^T) {}^b\mathbf{w}_{b/p} \quad (4.21)$$

soit en simplifiant :

$${}^{k-1}\mathbf{w}_k = {}^{k-1}\mathbf{R}_k {}^b\mathbf{R}_k^T {}^b\mathbf{w}_{b/p} \quad (4.22)$$

Finalement, l'expression du modèle cinématique inverse du module à l'envers est la suivante :

$$\dot{\mathbf{q}}_{ak} = {}^{k-1}\mathbf{J}_k^{-1} {}^{k-1}\mathbf{w}_k \quad (4.23)$$

avec la matrice jacobienne du module définie par :

$${}^{k-1}\mathbf{J}_k^{-1} = - {}^b\mathbf{J}_p^{-1} {}^b\mathbf{R}_k {}^{k-1}\mathbf{R}_k^T \quad (4.24)$$

On peut remarquer que le calcul de la matrice jacobienne inverse du module à l'envers peut se simplifier si on l'exprime dans le repère de la plate-forme Σ_k , soit :

$${}^k\mathbf{J}_k^{-1} = {}^{k-1}\mathbf{J}_k^{-1} {}^{k-1}\mathbf{R}_k = - {}^b\mathbf{J}_p^{-1} {}^b\mathbf{R}_k \quad (4.25)$$

Résumé pour le calcul du modèle cinématique inverse

Le calcul du modèle cinématique inverse d'un module du robot s'exprime sous la forme :

$$\dot{\mathbf{q}}_{ak} = {}^{k-1}\mathbf{J}_k^{-1} {}^{k-1}\mathbf{w}_k = {}^k\mathbf{J}_k^{-1} {}^k\mathbf{w}_k \quad (4.26)$$

Avec :

- Pour le calcul du MCI pour un module à l'endroit :
 1. Calculer ${}^b\mathbf{R}_p = {}^{k-1}\mathbf{R}_k$;
 2. Calculer ${}^b\mathbf{P}_{Ci} = {}^b\mathbf{R}_p {}^p\mathbf{P}_{Ci}$ pour $i=1,2$;
 3. Calculer ${}^{k-1}\mathbf{J}_k^{-1} = {}^b\mathbf{J}_p^{-1}$ en utilisant (4.16);
 4. Calculer ${}^k\mathbf{J}_k^{-1} = {}^{k-1}\mathbf{J}_k^{-1} {}^{k-1}\mathbf{R}_k$.
- Pour le calcul du MCI pour un module à l'envers :
 1. Calculer ${}^b\mathbf{R}_p$ en utilisant (4.5), puis calculer θ et ϕ à partir de (4.13) et (4.18);
 2. Calculer ${}^b\mathbf{P}_{Ci} = {}^b\mathbf{R}_k {}^{k-1}\mathbf{R}_k^T {}^{k-1}\mathbf{R}_p {}^p\mathbf{P}_{Ci}$ pour $i=1,2$;
 3. Calculer ${}^b\mathbf{J}_p^{-1}$ en utilisant (4.16);
 4. Calculer ${}^k\mathbf{J}_k^{-1} = -{}^b\mathbf{J}_p^{-1} {}^b\mathbf{R}_k$.

4.3.2 Modèle cinématique inverse du second ordre

Le modèle cinématique inverse du second ordre s'écrit dans le cas d'un module à l'endroit :

$$\ddot{\mathbf{q}}_{ak} = {}^k\dot{\mathbf{J}}_k^{-1} {}^k\mathbf{w}_k + {}^k\mathbf{J}_k^{-1} {}^k\dot{\mathbf{w}}_k \quad (4.27)$$

L'application de cette équation au cas du module à l'envers est assez simple, tous les calculs étant déjà faits :

- le calcul de ${}^k\mathbf{J}_k^{-1}$ se fait suivant l'algorithme présenté au paragraphe précédent ;
- le calcul de ${}^{k-1}\dot{\mathbf{J}}_k^{-1}$ se fait de la même manière :
 1. Calculer ${}^b\mathbf{R}_p$ en utilisant (4.5), puis calculer θ et ϕ à partir de (4.13) et (4.18);
 2. Calculer ${}^b\dot{\mathbf{J}}_p^{-1}$ en utilisant les résultats du chapitre 3;
 3. Calculer ${}^k\dot{\mathbf{J}}_k^{-1} = -{}^b\dot{\mathbf{J}}_p^{-1} {}^b\mathbf{R}_k$.

4.4 Modèle dynamique inverse de l'anguille

En prenant comme variables d'état du système les variables cartésiennes en position, vitesse et accélération des plates-formes, le modèle dynamique inverse permet de calculer les couples articulaires mais aussi les accélérations de la tête de l'anguille. On présente ici l'algorithme de Newton-Euler qui permet de calculer ces deux quantités.

4.4.1 Rappel : Modèle dynamique d'un module

Nous avons vu dans le chapitre 3 que le modèle dynamique d'un module parallèle peut s'écrire sous la forme suivante [Ibrahim, 2006] :

$$\mathbf{\Gamma} = {}^k\mathbf{J}_k^T {}^k\mathbf{M}_k + \sum_{i=1}^m \left(\frac{\partial \dot{\mathbf{q}}_{ik}}{\partial \dot{\mathbf{q}}_{ak}} \right)^T \mathbf{H}_i \quad (4.28)$$

avec :

- $\mathbf{\Gamma}$: le vecteur des couples articulaires ;
- \mathbf{M}_k : les moments des efforts extérieurs appliqués à la plate-forme ;
- \mathbf{J}_p : la matrice jacobienne (3×3) du module calculé précédemment ;
- m : le nombre de jambe du robot ;
- \mathbf{H}_i : le modèle dynamique inverse de la jambe i ;
- $\dot{\mathbf{q}}_{ik}$: le vecteur des vitesses articulaires de la jambe i du module k ;
- $\dot{\mathbf{q}}_{ak}$: le vecteur des vitesses articulaires motorisées du module k .

Dans le cas des modules utilisés, on a montré que les inerties des différentes jambes pouvaient être considérées comme négligeables comparées à celles de la base et de la plate-forme. Ainsi, le modèle dynamique d'un module s'écrit en tenant compte cette fois de l'inertie des moteurs ainsi que de leurs frottements secs et visqueux :

$$\mathbf{\Gamma} = {}^k\mathbf{J}_k^T {}^k\mathbf{M}_k + \mathbf{I}_{ak} \ddot{\mathbf{q}}_{ak} + \mathbf{F}_v \dot{\mathbf{q}}_{ak} + \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}_{ak}) \quad (4.29)$$

avec :

- \mathbf{I}_{ak} la matrice diagonale (3×3) représentant les inerties des actionneurs ;
- \mathbf{F}_v la matrice diagonale (3×3) représentant les frottements visqueux des actionneurs ;
- \mathbf{F}_s la matrice diagonale (3×3) représentant les frottements secs des actionneurs.

On peut également écrire le modèle dynamique sous forme réduite :

$$\boldsymbol{\tau} = {}^k\mathbf{J}_k^T {}^k\mathbf{M}_k + \mathbf{I}_{ak} \ddot{\mathbf{q}}_{ak} \quad (4.30)$$

$$\boldsymbol{\tau} = \mathbf{\Gamma} - \mathbf{F}_v \dot{\mathbf{q}}_{ak} - \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}_{ak}) \quad (4.31)$$

4.4.2 Modèle cinématique du robot

Jusqu'ici, on a calculé le modèle cinématique d'un seul module en supposant que la base de celui-ci était fixe. Dans le cas présent, la base de chaque module k est mobile et possède un torseur cinématique \mathbb{V}_{k-1} , avec :

$$\mathbb{V}_{k-1} = \begin{bmatrix} \mathbf{V}_{k-1}^T & \boldsymbol{\omega}_{k-1}^T \end{bmatrix}^T$$

avec :

- \mathbf{V}_{k-1} : vitesse linéaire de l'origine du repère Σ_{k-1} ;
- $\boldsymbol{\omega}_{k-1}$: vitesse angulaire du repère Σ_{k-1} .

Ainsi, connaissant la vitesse relative de la plate-forme du module k par rapport à sa base, on peut en déduire le torseur cinématique de la plate-forme soit :

$$\mathbb{V}_k = \mathbb{V}_{k-1} + \mathbf{a}_k \mathbf{w}_k \quad (4.32)$$

où : \mathbf{a}_k représente une matrice (6×3) de projection entre la vitesse relative du module et le torseur (6×6) cinématique relatif, telque :

$$\mathbf{v}_k = \mathbf{a}_k \mathbf{w}_k \quad (4.33)$$

Comme le module parallèle est assimilable à une articulation sphérique, la matrice \mathbf{a}_k s'écrit alors :

$$\mathbf{a}_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{d3} \end{bmatrix}^T \quad (4.34)$$

avec \mathbf{I}_{d3} la matrice identité (3×3) .

Pour pouvoir écrire les repères de projection dans l'équation de calcul du torseur cinématique (équation 4.32), on définit la matrice de transformation qui permet de passer du repère Σ_{k-1} au repère Σ_k du robot pour $k=1, \dots, n$:

$${}^{k-1}\mathbf{T}_k = \begin{bmatrix} {}^{k-1}\mathbf{R}_k & {}^{k-1}\mathbf{P}_k \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (4.35)$$

avec :

- ${}^{k-1}\mathbf{R}_k$ la matrice (3×3) de rotation entre la plate-forme et la base du module k ;
- ${}^{k-1}\mathbf{P}_k$ le vecteur (3×1) de position du repère Σ_k par rapport au repère Σ_{k-1} du module k et défini par :

$${}^{k-1}\mathbf{P}_k = \begin{bmatrix} d_k & 0 & 0 \end{bmatrix}^T$$

- d_k la distance entre les axes \mathbf{z}_{k-1} et \mathbf{z}_k le long de \mathbf{x}_{k-1} .

L'équation (4.32) du torseur cinématique (avec les repères de projection) s'écrit alors :

$${}^k\mathbb{V}_k = {}^k\mathbb{T}_{k-1} {}^{k-1}\mathbb{V}_{k-1} + {}^k\mathbf{v}_k \quad (4.36)$$

avec :

$${}^k\mathbb{T}_{k-1} = \begin{bmatrix} {}^k\mathbf{R}_{k-1} & -{}^k\mathbf{R}_{k-1} {}^{k-1}\hat{\mathbf{P}}_k \\ \mathbf{0}_{3 \times 3} & {}^k\mathbf{R}_{k-1} \end{bmatrix} \quad (4.37)$$

Pour initialiser la récurrence du modèle cinématique, on supposera connue à $t=0$ la matrice ${}^w\mathbf{T}_0$ et le torseur ${}^0\mathbb{V}_0$ désignant respectivement la matrice de transformation entre un repère terrestre fixe Σ_w et le repère de la tête Σ_0 , et le torseur cinématique de vitesse de la tête dans le repère Σ_0 . Ces deux quantités seront mises à jour à chaque pas de simulation par intégration numérique de l'accélération de la tête.

Pour les simulations, nos variables d'état sont les 3 angles de cardan $(\theta_k, \phi_k, \psi_k)$ ainsi que leurs dérivées. Les vitesses et accélérations absolues sont alors calculées par les relations suivantes pour $k=1, \dots, n$:

$${}^k\mathbb{T}_{k-1} = \begin{bmatrix} {}^k\mathbf{R}_{k-1} & -{}^k\mathbf{R}_{k-1} {}^{k-1}\hat{\mathbf{P}}_k \\ \mathbf{0}_{3 \times 3} & {}^k\mathbf{R}_{k-1} \end{bmatrix} \quad (4.38)$$

$${}^k\mathbf{w}_k = {}^k\boldsymbol{\Omega}_{RTL} \dot{\boldsymbol{\Theta}}_k \quad (4.39)$$

$${}^k\mathbb{V}_k = {}^k\mathbb{T}_{k-1} {}^{k-1}\mathbb{V}_{k-1} + {}^k\mathbf{a}_k {}^k\mathbf{w}_k \quad (4.40)$$

$${}^k\dot{\mathbf{w}}_k = {}^k\dot{\boldsymbol{\Omega}}_{RTL} \dot{\boldsymbol{\Theta}}_k + {}^k\boldsymbol{\Omega}_{RTL} \ddot{\boldsymbol{\Theta}}_k \quad (4.41)$$

$${}^k\gamma_k = {}^k\mathbf{a}_k {}^k\dot{\mathbf{w}}_k + {}^k\xi_k \quad (4.42)$$

$${}^k\xi_k = \begin{bmatrix} {}^k\mathbf{R}_{k-1} \left[{}^{k-1}\hat{\mathbf{w}}_{k-1} \left({}^{k-1}\hat{\mathbf{w}}_{k-1} {}^{k-1}\mathbf{P}_k \right) \right] \\ {}^k\hat{\mathbf{w}}_{k-1} {}^k\mathbf{w}_k \end{bmatrix} \quad (4.43)$$

$${}^k\dot{\mathbb{V}}_k = {}^k\mathbb{T}_{k-1} {}^{k-1}\dot{\mathbb{V}}_{k-1} + {}^k\gamma_k \quad (4.44)$$

avec :

- ${}^{k-1}\hat{\mathbf{P}}_k$: matrice (3×3) du pré-produit vectoriel associé à ${}^{k-1}\mathbf{P}_k$;
- ${}^k\boldsymbol{\Omega}_{RTL}$: matrice (3×3) permettant d'exprimer ${}^k\mathbf{w}_k$ en fonction de $\dot{\boldsymbol{\Theta}}_k$ et définie par :

$${}^k\boldsymbol{\Omega}_{RTL} = \begin{bmatrix} C\phi_k C\psi_k & S\psi_k & 0 \\ -C\phi_k S\psi_k & C\psi_k & 0 \\ S\phi_k & 0 & 1 \end{bmatrix} \quad (4.45)$$

4.4.3 Calcul du Modèle Dynamique Inverse de l'anguille

L'algorithme du modèle dynamique inverse présenté ici consiste en trois récurrences (une avant, une arrière puis une avant). Il est à noter ici que pour les robots à base fixe, seules les deux premières récurrences existent. Dans notre cas, une troisième est nécessaire car il faut déterminer l'accélération de la base (la tête de l'anguille) avant de pouvoir calculer les couples des actionneurs.

Première récurrence avant

Dans cette première étape, on calcule pour chaque module, la matrice de transformation entre torseurs ${}^k\mathbb{T}_{k-1}$, le torseur cinématique ainsi que tous les éléments d'accélération qui ne dépendent pas de l'accélération de la tête ($\dot{\mathbf{V}}_0, \dot{\boldsymbol{\omega}}_0$).

On calcule donc pour $k=1, \dots, n$: ${}^k\mathbb{T}_{k-1}$, ${}^k\mathbb{V}_k$ et ${}^k\boldsymbol{\gamma}_k$ en utilisant respectivement (4.38), (4.40) et (4.42). On calcule également pour $k=0, \dots, n$ le torseur ${}^k\boldsymbol{\beta}_k$ qui représente les éléments des équations de Newton-Euler qui ne dépendent pas des termes d'accélération du module ainsi que ${}^k\mathbf{f}_{ek}$ qui représente le torseur des efforts appliqués par le module k sur le fluide :

$${}^k\boldsymbol{\beta}_k = {}^k\mathbf{f}_{ek} + \begin{bmatrix} {}^k\boldsymbol{\omega}_k \times ({}^k\boldsymbol{\omega}_k \times {}^k\mathbf{MS}_k) \\ {}^k\boldsymbol{\omega}_k \times ({}^k\mathbf{I}_k {}^k\boldsymbol{\omega}_k) \end{bmatrix} \quad (4.46)$$

Récurrence arrière

Dans cette deuxième étape, on cherche à calculer l'accélération de la tête à partir des paramètres inertiels du corps composé de l'anguille, où le corps composé d'un module k consiste en la somme des paramètres inertiels des plates-formes des modules k, \dots, n . Pour rappel, la plate-forme du module $k-1$ constitue également la base du module k .

Pour pouvoir calculer le corps composé d'un module k , il faut commencer par écrire son équation d'équilibre mécanique. Lorsque les modules sont connectés en série, les forces permettant l'équilibre d'un module k sont représentées sur la figure (4.5) et l'équation associée s'exprime ainsi :

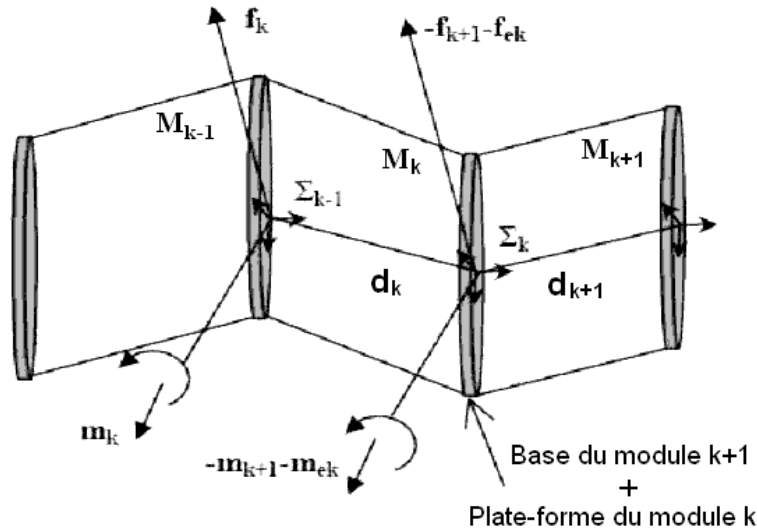


Figure 4.5: – Forces et moments exercés sur le module k

$${}^k\mathbf{f}_k = ({}^k\mathbb{F}_k + {}^k\mathbb{F}_{bk}) + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\mathbf{f}_{k+1} + {}^k\mathbf{f}_{ek} \quad (4.47)$$

où :

- \mathbf{f}_k est le torseur (6×1) des efforts exercés par le module $k-1$ sur le module k ;
- \mathbb{F}_{bk} représente la somme des torseurs d'efforts de tous les corps de toutes les jambes du module k . Comme jusqu'ici on a considéré que les inerties des jambes étaient négligeables, alors on considère que les efforts produits le sont de même soit : $\mathbb{F}_{bk} = 0$;
- \mathbb{F}_k représente la somme des efforts inertiels de la plate-forme. L'application des équations de Newton-Euler nous donne :

$$\mathbb{F}_k = \mathbb{J}_k \dot{\mathbf{V}}_k + \begin{bmatrix} \boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \mathbf{M}\mathbf{S}_k) \\ \boldsymbol{\omega}_k \times (\mathbf{I}_k \boldsymbol{\omega}_k) \end{bmatrix} \quad (4.48)$$

En utilisant les équations (4.48) et (4.46), l'équation d'équilibre (4.47) peut être réécrite de la manière suivante :

$${}^k\mathbf{f}_k = {}^k\mathbb{J}_k {}^k\dot{\mathbf{V}}_k + {}^k\boldsymbol{\beta}_k + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\mathbf{f}_{k+1} \quad (4.49)$$

En appliquant les équations de Newton-Euler sur le module composé k et en remarquant que ${}^{n+1}\mathbf{f}_{n+1} = 0$, l'équation (4.49) peut se réécrire comme :

$$\begin{aligned} {}^k\mathbf{f}_k &= {}^k\mathbb{J}_k {}^k\dot{\mathbf{V}}_k + {}^k\boldsymbol{\beta}_k + {}^{k+1}\mathbb{T}_k^T \left({}^{k+1}\mathbb{J}_{k+1} {}^{k+1}\dot{\mathbf{V}}_{k+1} + {}^{k+1}\boldsymbol{\beta}_{k+1} \right) + \dots \\ &+ {}^n\mathbb{T}_{n-1}^T \left({}^n\mathbb{J}_n {}^n\dot{\mathbf{V}}_n + {}^n\boldsymbol{\beta}_n \right) \end{aligned} \quad (4.50)$$

En substituant ${}^{k+1}\dot{\mathbf{V}}_{k+1}, \dots, {}^n\dot{\mathbf{V}}_n$ en fonction de ${}^k\dot{\mathbf{V}}_k$ en utilisant (4.44), on obtient :

$${}^{k+1}\dot{\mathbf{V}}_{k+1} = {}^{k+1}\mathbb{T}_k {}^k\dot{\mathbf{V}}_k + {}^{k+1}\boldsymbol{\gamma}_{k+1}, \quad (4.51)$$

$$\begin{aligned} {}^{k+2}\dot{\mathbf{V}}_{k+2} &= {}^{k+2}\mathbb{T}_{k+1} {}^{k+1}\dot{\mathbf{V}}_{k+1} + {}^{k+2}\boldsymbol{\gamma}_{k+2} \\ &= {}^{k+2}\mathbb{T}_{k+1} \left({}^{k+1}\mathbb{T}_k {}^k\dot{\mathbf{V}}_k + {}^{k+1}\boldsymbol{\gamma}_{k+1} \right) + {}^{k+2}\boldsymbol{\gamma}_{k+2}, \end{aligned} \quad (4.52)$$

$${}^n\dot{\mathbf{V}}_n = {}^n\mathbb{T}_k {}^k\dot{\mathbf{V}}_k + {}^n\mathbb{T}_{k+1} {}^{k+1}\boldsymbol{\gamma}_{k+1} + \dots + {}^n\boldsymbol{\gamma}_n \quad (4.53)$$

A partir de (4.50), (4.52) et (4.53), on peut en déduire :

$${}^k\mathbf{f}_k = {}^k\mathbb{J}_k^c {}^k\dot{\mathbf{V}}_k + {}^k\boldsymbol{\beta}_k^c \quad (4.54)$$

avec :

$${}^k\mathbb{J}_k^c = {}^k\mathbb{J}_k + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\mathbb{J}_{k+1}^c {}^{k+1}\mathbb{T}_k \quad (4.55)$$

$${}^k\boldsymbol{\beta}_k^c = {}^k\boldsymbol{\beta}_k + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\boldsymbol{\beta}_{k+1}^c + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\mathbb{J}_{k+1}^c {}^{k+1}\boldsymbol{\gamma}_{k+1} \quad (4.56)$$

${}^k\mathbb{J}_k^c$ et ${}^k\beta_k^c$ représentent respectivement la matrice spatiale d'inertie et le torseur d'effort du module composé k.

Pour $k=0$, en remarquant que ${}^0\mathbf{f}_0$ est égal à zéro, on obtient donc en utilisant (4.54) l'équation permettant de calculer l'accélération de la tête de l'anguille :

$${}^0\dot{\mathbf{V}}_0 = - \left({}^0\mathbb{J}_0^c \right)^{-1} {}^0\beta_0^c \quad (4.57)$$

Pour résumer, la récurrence arrière s'opère en initialisant ${}^n\mathbb{J}_n^c = {}^n\mathbb{J}_n$ et ${}^n\beta_n^c = {}^n\beta_n$ puis pour $k=n-1, \dots, 0$ on calcule les deux équations (4.55) et (4.56). A la fin, l'accélération de la tête ${}^0\dot{\mathbf{V}}_0$ est obtenue par la relation (4.57).

Deuxième récurrence avant

Dans cette dernière étape, on calcule l'accélération absolue du module k à partir de l'accélération de la tête obtenue dans la récurrence précédente. On calcule également pour $k=1, \dots, n$ le torseur d'efforts ${}^k\mathbf{f}_k$ pour finir par le calcul des couples moteurs :

$${}^k\dot{\mathbf{V}}_k = {}^k\mathbb{T}_{k-1} {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\gamma_k \quad (4.58)$$

$${}^k\mathbf{f}_k = {}^k\mathbb{J}_k^c {}^k\dot{\mathbf{V}}_k + {}^k\beta_k^c \quad (4.59)$$

A partir de l'équation (4.29), le calcul des couples moteurs se fait en sélectionnant la partie correspondant aux moments de ${}^k\mathbf{f}_k$ (en utilisant \mathbf{a}_k) puis en projetant le résultat sur l'espace articulaire en utilisant la matrice jacobienne du module. En prenant en compte les paramètres de frottements secs (\mathbf{F}_{sk}), les frottements visqueux (\mathbf{F}_{vk}) ainsi que les inerties des rotors des moteurs (\mathbf{I}_{ak}), le vecteur des couples moteurs est donnée par :

$$\mathbf{\Gamma}_k = {}^k\mathbf{J}_k^T {}^k\mathbf{a}_k^T {}^k\mathbf{f}_k + \mathbf{F}_{sk} \text{sign}(\dot{\mathbf{q}}_{ak}) + \mathbf{F}_{vk} \dot{\mathbf{q}}_{ak} + \mathbf{I}_{ak} \ddot{\mathbf{q}}_{ak} \quad (4.60)$$

où : \mathbf{F}_{sk} , \mathbf{F}_{vk} et \mathbf{I}_{ak} sont des matrices diagonales (3×3), avec :

- $\mathbf{F}_{sk} = \text{diag} (F_{sk1}, F_{sk2}, F_{sk3})$, la matrice des frottements secs des articulations motorisées ;
- $\mathbf{F}_{vk} = \text{diag} (F_{vk1}, F_{vk2}, F_{vk3})$, la matrice des frottements visqueux des articulations motorisées ;
- $\mathbf{I}_{ak} = \text{diag} (I_{ak1}, I_{ak2}, I_{ak3})$, la matrice des inerties des articulations motorisées.

4.5 Modèle dynamique direct de l'anguille

Le modèle dynamique direct permet de calculer les accélérations cartésiennes des plates-formes ainsi que l'accélération absolue de la tête de l'anguille en fonction des positions et vitesses cartésiennes des plates-formes ainsi que des couples moteurs. Dans la suite, on suppose que les positions et vitesses articulaires sont connues et calculées par les modèles géométriques et cinématiques inverses.

4.5.1 Préliminaires

Avant de pouvoir calculer le modèle dynamique direct de l'anguille, on essaye dans cette étape préliminaire d'exprimer l'accélération relative du module k ${}^k\dot{\mathbf{w}}_k$ en fonction de l'accélération absolue du module précédent ${}^{k-1}\dot{\mathbf{V}}_{k-1}$. Cette relation s'obtient en exprimant le modèle cinématique inverse du second ordre ainsi que de l'expression des couples actionneurs :

– Le modèle cinématique inverse du second ordre d'un module est donné par :

$$\ddot{\mathbf{q}}_{ak} = {}^k\mathbf{J}_k^{-1} \left({}^k\dot{\mathbf{w}}_k - {}^k\dot{\mathbf{J}}_k \dot{\mathbf{q}}_{ak} \right) \quad (4.61)$$

– Le calcul du vecteur des couples actionneurs pour un seul module est donné par (4.29) :

$$\boldsymbol{\tau}_k = {}^k\mathbf{J}_k^T {}^k\mathbf{a}_k^T \left({}^k\mathbb{F}_k + {}^k\mathbf{f}_{ek} \right) + \mathbf{I}_{ak} \ddot{\mathbf{q}}_{ak} \quad (4.62)$$

$$\boldsymbol{\tau}_k = \boldsymbol{\Gamma}_k - \mathbf{F}_{sk} \text{sign}(\dot{\mathbf{q}}_{ak}) - \mathbf{F}_{vk} \dot{\mathbf{q}}_{ak} \quad (4.63)$$

En substituant (4.61) dans (4.62), on obtient alors une expression du vecteur des couples ne faisant plus intervenir les accélérations articulaires :

$$\boldsymbol{\tau}_k = {}^k\mathbf{J}_k^T {}^k\mathbf{a}_k^T \left({}^k\mathbb{F}_k + {}^k\mathbf{f}_{ek} \right) + \mathbf{I}_{ak} {}^k\mathbf{J}_k^{-1} \left({}^k\dot{\mathbf{w}}_k - {}^k\dot{\mathbf{J}}_k \dot{\mathbf{q}}_{ak} \right) \quad (4.64)$$

A partir de (4.48), (4.46) puis (4.40), on pose alors :

$${}^k\mathbb{F}_k + {}^k\mathbf{f}_{ek} = {}^k\mathbb{J}_k {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\boldsymbol{\beta}_k = {}^k\mathbb{J}_k \left({}^k\mathbb{T}_{k-1} {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\mathbf{a}_k {}^k\dot{\mathbf{w}}_k + {}^k\boldsymbol{\xi}_k \right) + {}^k\boldsymbol{\beta}_k \quad (4.65)$$

En substituant (4.65) dans (4.64) et en multipliant chacun des membres de l'équation par ${}^k\mathbf{J}_k^{-T}$, on obtient :

$${}^k\mathbf{J}_k^{-T} \boldsymbol{\tau}_k = \mathbf{A}_{tk} {}^k\dot{\mathbf{w}}_k + {}^k\mathbb{J}_{bk} {}^{k-1}\dot{\mathbf{V}}_{k-1} + \mathbf{H}_{tk} \quad (4.66)$$

avec :

$$\mathbf{A}_{tk} = {}^k\mathbf{a}_k^T {}^k\mathbb{J}_k {}^k\mathbf{a}_k + {}^k\mathbf{J}_k^{-T} \mathbf{I}_{ak} {}^k\mathbf{J}_k^{-1} \quad (4.67)$$

$${}^k\mathbb{J}_{bk} = {}^k\mathbf{a}_k^T {}^k\mathbb{J}_k {}^k\mathbb{T}_{k-1} \quad (4.68)$$

$$\mathbf{H}_{tk} = {}^k\mathbf{a}_k^T \left({}^k\mathbb{J}_k {}^k\boldsymbol{\xi}_k + {}^k\boldsymbol{\beta}_k \right) - {}^k\mathbf{J}_k^{-T} \mathbf{I}_{ak} {}^k\mathbf{J}_k^{-1} {}^k\dot{\mathbf{J}}_k \dot{\mathbf{q}}_{ak} \quad (4.69)$$

Finalement, on obtient la relation souhaitée :

$${}^k\dot{\mathbf{w}}_k = \mathbf{A}_{tk}^{-1} \left({}^k\mathbf{J}_k^{-T} \boldsymbol{\tau}_k - {}^k\mathbb{J}_{bk} {}^{k-1}\dot{\mathbf{V}}_{k-1} - \mathbf{H}_{tk} \right) \quad (4.70)$$

4.5.2 Calcul du Modèle Dynamique Direct de l'anguille

Comme pour le calcul du modèle dynamique inverse, l'algorithme présenté ici consiste en trois récurrences (une avant, une arrière puis une avant). Pour rappel, on cherche à déterminer les accélérations cartésiennes des plates-formes des modules, ainsi que le torseur d'accélération de la tête en fonction des positions et vitesses des plates-formes ainsi que des couples moteurs.

Première récurrence avant

Dans cette première étape, on calcule pour chaque module la matrice de transformation entre torseurs, le torseur cinématique ainsi que tous les éléments d'accélération qui ne dépendent pas de l'accélération de la tête ($\mathbf{V}_0, \boldsymbol{\omega}_0$) et des termes d'accélération relative ($\dot{\mathbf{w}}_k$).

On calcule donc pour $k=1, \dots, n$: ${}^k\mathbb{T}_{k-1}$, ${}^k\mathbb{V}_k$ et ${}^k\xi_k$ en utilisant respectivement (4.38), (4.40) et (4.43). On calcule également pour $k=0, \dots, n$ le torseur ${}^k\beta_k$ qui représente les éléments des équations de Newton-Euler (4.48) qui ne dépendent pas des termes d'accélération du module, soit pour rappel :

$$\begin{aligned} {}^k\xi_k &= \begin{bmatrix} {}^k\mathbf{R}_{k-1} \left[{}^{k-1}\hat{\boldsymbol{\omega}}_{k-1} \left({}^{k-1}\hat{\boldsymbol{\omega}}_{k-1} {}^{k-1}\mathbf{P}_k \right) \right] \\ {}^k\hat{\boldsymbol{\omega}}_{k-1} {}^k\mathbf{w}_k \end{bmatrix} \\ {}^k\beta_k &= {}^k\mathbf{f}_{ek} + \begin{bmatrix} {}^k\hat{\boldsymbol{\omega}}_k \left({}^k\hat{\boldsymbol{\omega}}_k {}^k\mathbf{M}\mathbf{S}_k \right) \\ {}^k\hat{\boldsymbol{\omega}}_k \left({}^k\mathbf{I}_k {}^k\boldsymbol{\omega}_k \right) \end{bmatrix} \end{aligned}$$

Récurrence arrière

Dans cette deuxième étape, on cherche à calculer l'accélération de la tête de l'anguille. A partir des résultats obtenus dans l'étape préliminaire, on cherche à exprimer le torseur des efforts ${}^k\mathbf{f}_k$ en fonction du torseur cinématique ${}^{k-1}\dot{\mathbf{V}}_{k-1}$. Pour ceci, on substitue l'équation (4.65) dans l'équation d'équilibre (4.47) qui s'écrit alors :

$${}^k\mathbf{f}_k = {}^k\mathbb{J}_k \left({}^k\mathbb{T}_{k-1} {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\mathbf{a}_k {}^k\dot{\mathbf{w}}_k + {}^k\xi_k \right) + {}^k\beta_k + {}^{k+1}\mathbb{T}_k^T {}^{k+1}\mathbf{f}_{k+1} \quad (4.71)$$

Or, à partir de (4.70) et sachant que pour $k=n$, ${}^{n+1}\mathbf{f}_{n+1} = 0$, cela nous permet d'exprimer directement ${}^n\mathbf{f}_n$ en fonction de ${}^{n-1}\dot{\mathbf{V}}_{n-1}$:

$${}^n\mathbf{f}_n = {}^n\mathbb{J}_n \left({}^n\mathbb{T}_{n-1} {}^{n-1}\dot{\mathbf{V}}_{n-1} + {}^n\mathbf{a}_n {}^n\dot{\mathbf{w}}_n + {}^n\xi_n \right) + {}^n\beta_n \quad (4.72)$$

En inversant l'équation (4.66), on obtient l'expression de ${}^n\dot{\mathbf{w}}_n$ que l'on peut introduire dans (4.72) :

$${}^n\mathbf{f}_n = {}^n\mathbb{K}_{n-1} {}^{n-1}\dot{\mathbf{V}}_{n-1} + {}^n\boldsymbol{\alpha}_n \quad (4.73)$$

avec :

$${}^n\mathbb{K}_{n-1} = {}^n\mathbb{J}_n {}^n\mathbb{T}_{n-1} - {}^n\mathbb{J}_n {}^n\mathbf{a}_n \mathbf{A}_{tn}^{-1} {}^n\mathbb{J}_{bn} \quad (4.74)$$

$${}^n\boldsymbol{\alpha}_n = {}^n\mathbb{J}_n {}^n\mathbf{a}_n \mathbf{A}_{tn}^{-1} \left({}^n\mathbf{J}_n^T \boldsymbol{\tau}_n - \mathbf{H}_{tn} \right) + {}^n\mathbb{J}_n {}^n\xi_n + {}^n\beta_n \quad (4.75)$$

Pour $k=n-1$, en remplaçant ${}^n\mathbf{f}_n$ dans (4.71) par l'expression définie dans (4.73), on a alors :

$$\begin{aligned} {}^{n-1}\mathbf{f}_{n-1} &= \left({}^{n-1}\mathbb{J}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\mathbb{K}_{n-1} \right) \left({}^{n-1}\mathbb{T}_{n-2} {}^{n-2}\dot{\mathbf{V}}_{n-2} + {}^{n-1}\mathbf{a}_{n-1} {}^{n-1}\dot{\mathbf{w}}_{n-1} + {}^{n-1}\boldsymbol{\xi}_{n-1} \right) \\ &+ \left({}^{n-1}\boldsymbol{\beta}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\boldsymbol{\alpha}_n \right) \end{aligned} \quad (4.76)$$

Par identification avec l'équation (4.73), on peut alors poser les termes de la récurrence :

$${}^{n-1}\mathbb{J}_{n-1}^* = {}^{n-1}\mathbb{J}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\mathbb{K}_{n-1} \quad (4.77)$$

$${}^{n-1}\boldsymbol{\beta}_{n-1}^* = {}^{n-1}\boldsymbol{\beta}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\boldsymbol{\alpha}_n \quad (4.78)$$

Finalement, pour $k=0$, sachant que ${}^0\mathbf{f}_0 = 0$, on peut en déduire que :

$${}^0\mathbf{f}_0 = 0 = {}^0\mathbb{J}_0 {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0 + {}^1\mathbf{f}_1 \quad (4.79)$$

soit :

$${}^0\mathbb{J}_0^* {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0^* = \mathbf{0} \quad (4.80)$$

On obtient alors l'accélération de la tête :

$${}^0\dot{\mathbf{V}}_0 = - \left({}^0\mathbb{J}_0^* \right)^{-1} {}^0\boldsymbol{\beta}_0^* \quad (4.81)$$

En résumé, cette récurrence consiste à initialiser ${}^n\mathbb{J}_n^* = {}^n\mathbb{J}_n$ et ${}^n\boldsymbol{\beta}_n^* = {}^n\boldsymbol{\beta}_n$ puis calculer pour $k=n, \dots, 1$:

$${}^k\mathbb{J}_{bk} = {}^k\mathbf{a}_k^T {}^k\mathbb{J}_k^* {}^k\mathbb{T}_{k-1} \quad (4.82)$$

$$\mathbf{A}_{tk} = {}^k\mathbf{a}_k^T {}^k\mathbb{J}_k^* {}^k\mathbf{a}_k + {}^k\mathbf{J}_k^{-T} \mathbf{I}_{ak} {}^k\mathbf{J}_k^{-1} \quad (4.83)$$

$$\mathbf{H}_{tk} = {}^k\mathbf{a}_k^T \left({}^k\mathbb{J}_k^* {}^k\boldsymbol{\xi}_k + {}^k\boldsymbol{\beta}_k^* \right) - {}^k\mathbf{J}_k^{-T} \mathbf{I}_{ak} {}^k\mathbf{J}_k^{-1} {}^k\dot{\mathbf{J}}_k \dot{\mathbf{q}}_{ak} \quad (4.84)$$

$${}^k\mathbb{K}_{k-1} = {}^k\mathbb{J}_k^* {}^k\mathbb{T}_{k-1} - {}^k\mathbb{J}_k^* {}^k\mathbf{a}_k \mathbf{A}_{tk}^{-1} {}^k\mathbb{J}_{bk} \quad (4.85)$$

$${}^k\boldsymbol{\alpha}_k = {}^k\mathbb{J}_k^* {}^k\mathbf{a}_k \mathbf{A}_{tk}^{-1} \left({}^k\mathbf{J}_k^{-T} \boldsymbol{\tau}_k - \mathbf{H}_{tk} \right) + {}^k\mathbb{J}_k^* {}^k\boldsymbol{\xi}_k + {}^k\boldsymbol{\beta}_k^* \quad (4.86)$$

$${}^{k-1}\mathbb{J}_{k-1}^* = {}^{k-1}\mathbb{J}_{k-1} + {}^k\mathbb{T}_{k-1}^T {}^k\mathbb{K}_{k-1} \quad (4.87)$$

$${}^{k-1}\boldsymbol{\beta}_{k-1}^* = {}^{k-1}\boldsymbol{\beta}_{k-1} + {}^k\mathbb{T}_{k-1}^T {}^k\boldsymbol{\alpha}_k \quad (4.88)$$

pour terminer par le calcul de ${}^0\dot{\mathbf{V}}_0$ en utilisant (4.81).

Deuxième récurrence avant

Dans cette dernière récurrence, on calcule les accélérations cartésiennes (${}^k\dot{\mathbf{w}}_k$) de la plateforme du module k , le torseur des efforts appliqués sur le module k (${}^k\mathbf{f}_k$) ainsi que le torseur des accélérations absolues (${}^k\dot{\mathbf{V}}_k$). Pour $k=1, \dots, n$ on calcule donc :

$${}^k\dot{\mathbf{w}}_k = \mathbf{A}_{tk}^{-1} \left({}^k\mathbf{J}_k^{-T} \boldsymbol{\tau}_k - {}^k\mathbb{J}_{bk} {}^{k-1}\dot{\mathbf{V}}_{k-1} - \mathbf{H}_{tk} \right) \quad (4.89)$$

$${}^k\mathbf{f}_k = {}^k\mathbb{K}_{k-1} {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\boldsymbol{\alpha}_k \quad (4.90)$$

$${}^k\dot{\mathbf{V}}_k = {}^k\mathbb{T}_{k-1} {}^{k-1}\dot{\mathbf{V}}_{k-1} + {}^k\mathbf{a}_k {}^k\dot{\mathbf{w}}_k + {}^k\boldsymbol{\xi}_k \quad (4.91)$$

4.6 Simulations

Dans ce paragraphe, on présente quelques résultats de simulation obtenus en utilisant les modèles dynamiques que nous avons étudiés. Ces simulations sont effectuées pour un robot composé de 12 modules.

4.6.1 Générateur de mouvements

Pour pouvoir simuler la nage de l'anguille, on utilise le même type de générateur de mouvements que celui décrit dans le chapitre 2. La structure du robot n'étant plus la même dans ce chapitre, on réécrit les équations de ce générateur de mouvements qui est composé de 3 fonctions ayant les rôles suivants :

1° Générateur de propulsion :

La propulsion de l'anguille est produite en utilisant une fonction sinusoïdale continue qu'on applique sur les angles ψ_k ($k=1,\dots,12$) du robot :

$$\psi(s, t) = f_1(t) \cdot A \cdot e^{\alpha s} \cdot \sin \left[2\pi \left(\frac{s}{\lambda} - \frac{t}{T} \right) \right] \quad (4.92)$$

pour laquelle :

- A : amplitude du mouvement ;
- α : réglage de l'exponentielle : augmentation de l'amplitude de la tête vers la queue ;
- T : période de l'oscillation ;
- λ : longueur d'onde de l'oscillation ;
- s : abscisse curviligne le long de la fibre neutre de l'anguille ;
- $f_1(t)$: fonction de lissage du démarrage et de l'arrêt ;

Pour appliquer cette loi de mouvements continue, on est obligé de discrétiser cette équation pour obtenir les valeurs à appliquer sur les angles ψ_k du robot [Yu *et al.*, 2005].

$$\psi_k(t) = \psi(X_{k+1}, t) - \psi(X_k, t) + f_2(t) O_{ff1} \quad (4.93)$$

avec :

- $X_k = \sum_{j=1}^k d_j$: la distance entre l'origine du repère Σ_k et le repère Σ_0 de la tête.
- $f_2(t)$: fonction pour amorcer les virages ;
- O_{ff1} : "offset" pour régler la courbure de l'anguille et ainsi faire des virages.

2° Réglage de l'altitude :

Pour que l'anguille puisse changer d'altitude (ou de profondeur) pendant sa nage, on impose sur les angles ϕ_k pour $k=1,\dots,12$ la fonction qui suit :

$$\phi_k = f_3(t) \cdot O_{ffs2} \quad (4.94)$$

On se contente dans l'état actuel des simulations d'imposer un "offset" O_{ffs2} constant sur les articulations pour faire des trajectoires de type looping ou plongée. Cet "offset" peut être ajusté par une fonction de démarrage et d'arrêt $f_3(t)$ semblable à $f_1(t)$.

3° Réglage du Roulis :

Dans le cadre des simulations effectuées dans ce manuscrit, nous avons décidé de ne pas utiliser les articulations de roulis. Par conséquent, on impose sur les angles θ_k pour $k=1,\dots,12$:

$$\theta_k = 0 \quad (4.95)$$

Les fonctions $f_1(t)$ à $f_3(t)$ sont des polynômes de degré 5 dont les coefficients respectent les conditions évoquées au chapitre 2.

4.6.2 Modèle d'interaction fluide-structure

Pour pouvoir simuler la nage du prototype, nous avons utilisé le même modèle fluide que celui présenté dans le chapitre 2. Pour ce modèle, nous avons fait l'hypothèse que le déplacement 3D du fluide autour du corps peut être modélisé pour chaque tranche de l'anguille comme un écoulement plan transverse (figure 4.6) :

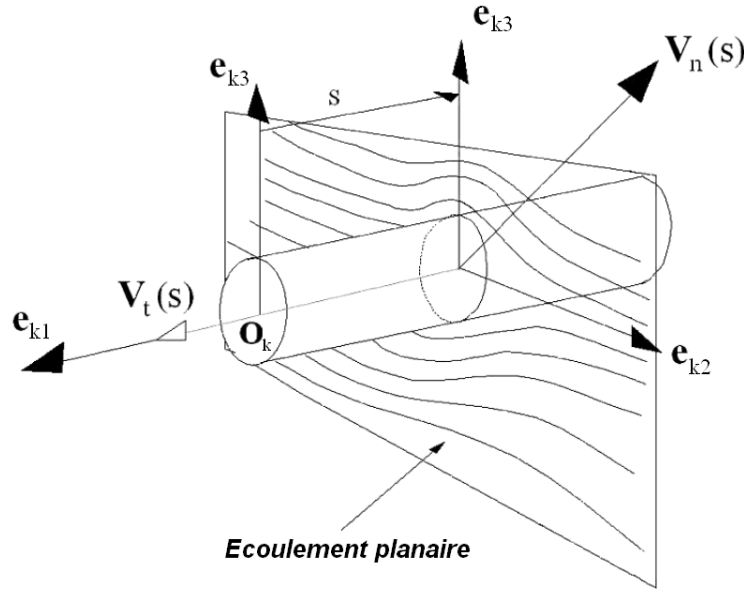


Figure 4.6: – Décomposition des vitesses dans le plan transverse d'une tranche de l'anguille

Dans ce cas de figure, les efforts fluides sont modélisés comme un champ d'efforts appliqués à chaque tranche de l'anguille et dépendant uniquement des mouvements de celle-ci. Pour chaque tranche d'un corps C_k de l'anguille, on définit un repère local $(\mathbf{e}_{k1}, \mathbf{e}_{k2}, \mathbf{e}_{k3})$ (voir figure 4.6) tel que :

$$\mathbf{e}_{k1} = \mathbf{x}_k, \mathbf{e}_{k2} = \mathbf{y}_k, \mathbf{e}_{k3} = \mathbf{z}_k$$

Pour chaque tranche s de ce corps de l'anguille, on calcule le torseur cinématique ${}^s\mathbb{V}_k$ au centre de la section en utilisant la relation suivante :

$${}^s\mathbb{V}_k = {}^s\mathbb{T}_k {}^k\mathbb{V}_k = \begin{bmatrix} {}^s\mathbf{V}_k & {}^s\boldsymbol{\omega}_k \end{bmatrix}^T \quad (4.96)$$

avec :

$${}^s\mathbb{T}_k = \begin{bmatrix} \mathbf{I}_{d3} & -{}^k\hat{\mathbf{P}}_s \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{d3} \end{bmatrix} \quad (4.97)$$

et

$${}^k\mathbf{P}_s = \begin{bmatrix} s & 0 & 0 \end{bmatrix}^T \quad (4.98)$$

On décompose alors les vitesses ${}^s\mathbf{V}_k$ et ${}^s\boldsymbol{\omega}_k$ dans le repère de $(\mathbf{e}_{k1}, \mathbf{e}_{k2}, \mathbf{e}_{k3})$ par :

$${}^s\mathbf{V}_k = V_{tk}\mathbf{e}_{k1} + V_{nk2}\mathbf{e}_{k2} + V_{nk3}\mathbf{e}_{k3} \quad (4.99)$$

$${}^s\boldsymbol{\omega}_k = \boldsymbol{\omega}_{tk}\mathbf{e}_{k1} + \boldsymbol{\omega}_{nk2}\mathbf{e}_{k2} + \boldsymbol{\omega}_{nk3}\mathbf{e}_{k3} \quad (4.100)$$

Puis on définit :

$$\|\mathbf{V}_{nk}\| = \sqrt{V_{nk2}^2 + V_{nk3}^2} \quad (4.101)$$

En utilisant les résultats du chapitre 2, le modèle de contact entre le fluide et les corps de l'anguille (modèle de Morison [Morison *et al.*, 1950]) peut s'écrire sous la forme ci-dessous :

$$\begin{aligned} {}^k\mathbf{f}_{hk} = & \begin{bmatrix} {}^k\mathbf{m}_{aGk} & -{}^k\mathbf{m}_{aGk} {}^k\hat{\mathbf{S}}_k \\ {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk} & {}^k\mathbf{I}_{aGk} - {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk} {}^k\hat{\mathbf{S}}_k \end{bmatrix} {}^k\dot{\mathbf{V}}_k + \\ & \left[\begin{pmatrix} {}^k\mathbf{m}_{aGk} ({}^k\hat{\boldsymbol{\omega}}_k {}^k\hat{\boldsymbol{\omega}}_k {}^k\mathbf{S}_k) \\ \left(\frac{4}{3} {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk}\right) ({}^k\hat{\boldsymbol{\omega}}_k {}^k\hat{\boldsymbol{\omega}}_k {}^k\mathbf{S}_k) \end{pmatrix} \right] + \int_0^{d_{k+1}} {}^s\mathbb{T}_k^T {}^s\mathbf{f}_{\text{drag},k} ds \end{aligned} \quad (4.102)$$

avec :

$${}^k\mathbf{m}_{aGk} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{lm2,k}d_{k+1} & 0 \\ 0 & 0 & C_{lm3,k}d_{k+1} \end{bmatrix} \quad {}^k\mathbf{I}_{aGk} = \begin{bmatrix} C_{am1,k}d_{k+1} & 0 & 0 \\ 0 & \frac{C_{lm3,k}d_{k+1}^3}{12} & 0 \\ 0 & 0 & \frac{C_{lm2,k}d_{k+1}^3}{12} \end{bmatrix}$$

et :

$${}^k\mathbf{f}_{\text{drag}}(s) = C_{ld1,k} |V_{tk}(s)| V_{tk}(s) \mathbf{e}_{k1} + \sum_{i=2}^3 C_{ldi,k} \|V_{nk}(s)\| V_{nki}(s) \mathbf{e}_{ki} \quad (4.103)$$

$${}^k\mathbf{m}_{\text{drag}}(s) = C_{ad1,k} |\boldsymbol{\omega}_{tk}| \boldsymbol{\omega}_{tk} \mathbf{e}_{k1} \quad (4.104)$$

Les termes $C_{ldi,k}$, $C_{lmi,k}$, $C_{ad1,k}$ et $C_{am1,k}$ sont des coefficients constants qui dépendent de la géométrie des sections de l'anguille (ici elliptiques), de la masse volumique de l'eau et du nombre de Reynolds (voir chapitre 2).

Dans l'équation (4.102), le terme dépendant du torseur des accélérations du corps C_k , que l'on appelle masse ajoutée, va venir s'additionner à la matrice spatiale d'inertie ${}^k\mathbb{J}_k$, tandis que l'autre terme constitue le torseur des efforts extérieurs ${}^k\mathbf{f}_{ek}$:

$${}^k\mathbb{J}_k = {}^k\mathbb{J}_k + {}^k\mathbb{J}_k^{\text{aj}} \quad (4.105)$$

avec :

$${}^k\mathbb{J}_k^{aj} = \begin{bmatrix} {}^k\mathbf{m}_{aGk} & -{}^k\mathbf{m}_{aGk} {}^k\hat{\mathbf{S}}_k \\ {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk} & {}^k\mathbf{I}_{aGk} - {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk} {}^k\hat{\mathbf{S}}_k \end{bmatrix} \quad (4.106)$$

et :

$${}^k\mathbf{f}_{ek} = \begin{bmatrix} {}^k\mathbf{m}_{aGk} ({}^k\hat{\boldsymbol{\omega}}_k {}^k\hat{\boldsymbol{\omega}}_k {}^k\mathbf{S}_k) \\ \left(\frac{4}{3} {}^k\hat{\mathbf{S}}_k {}^k\mathbf{m}_{aGk}\right) ({}^k\hat{\boldsymbol{\omega}}_k {}^k\hat{\boldsymbol{\omega}}_k {}^k\mathbf{S}_k) \end{bmatrix} + \int_0^{d_{k+1}} {}^s\mathbb{T}_k^T {}^s\mathbf{f}_{drag,k} ds \quad (4.107)$$

Les matrices ${}^k\mathbb{J}_k^{aj}$ et ${}^k\mathbf{f}_{ek}$ ne dépendant pas des termes d'accélération, elles seront calculées durant la première récurrence avant de l'algorithme de Newton-Euler (pour les modèles dynamiques inverse et direct).

4.6.3 Validation des algorithmes

Validation croisée des modèles dynamiques direct et inverse

Dans ce paragraphe, on cherche à valider les algorithmes des modèles dynamiques inverse et direct. Pour effectuer cette validation croisée, une simulation est effectuée en utilisant le modèle dynamique inverse pour laquelle on spécifie $\boldsymbol{\Theta}(t)$, $\dot{\boldsymbol{\Theta}}(t)$, $\ddot{\boldsymbol{\Theta}}(t)$ comme entrées et ${}^0\mathbf{V}_0 = 0$, $\boldsymbol{\Sigma}_0 \equiv \boldsymbol{\Sigma}_w$ comme conditions initiales. Ensuite, on simule le modèle dynamique direct en utilisant comme entrées $\boldsymbol{\Theta}(t)$, $\dot{\boldsymbol{\Theta}}(t)$, $\boldsymbol{\Gamma}(t)$ obtenues par le modèle dynamique inverse et les mêmes conditions initiales. Une fois ces deux simulations effectuées, on doit vérifier que :

- le calcul de l'accélération de la tête est le même que ce soit en utilisant le modèle dynamique inverse ou le modèle dynamique direct ;
- les accélérations relatives $\dot{\mathbf{w}}_k$ des plate-formes calculées par le modèle dynamique direct sont les mêmes que les accélérations relatives imposées par le générateur de mouvements pour le modèle dynamique inverse.

Pour effectuer les simulations, on impose la loi articulaire proposée au paragraphe précédent sur les 3 angles θ_k , ϕ_k et ψ_k de chaque module. Les paramètres utilisés pour les simulations à venir sont les suivants :

$$A=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; t_f=3 \text{ s} ; \lambda=1,5 \text{ m} ; T=3 \text{ s} ; O_{ff_1}=0 ; O_{ff_2}=0 ; O_{ff_3}=0$$

Le schéma Simulink permettant de faire cette validation est donné sur la figure (4.7) :

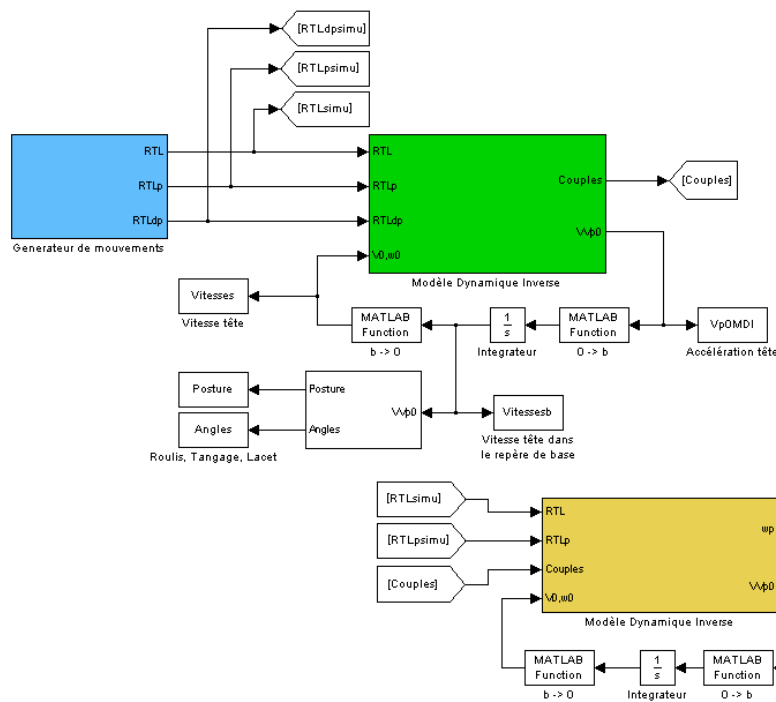


Figure 4.7: – Simulateur pour la validation des algorithmes

A partir de cette loi articulaire, les figures (4.8 -a) et (4.8 -b) montrent respectivement la comparaison de l'accélération de la tête (${}^0\dot{\mathbf{V}}_0$) calculée avec les deux algorithmes et la comparaison entre les accélérations cartésiennes ($\dot{\mathbf{w}}_k$) calculées par le modèle dynamique inverse et les accélérations imposées au modèle dynamique inverse :

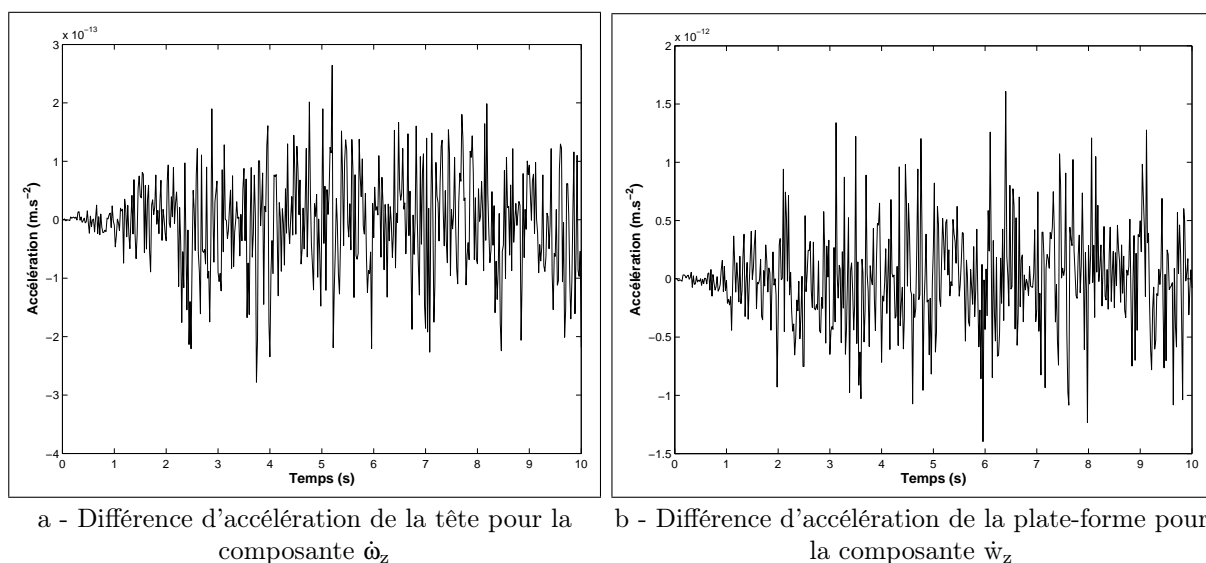


Figure 4.8: – Différence des accélérations entre le MDI et le MDD

Sur ces deux figures, on remarque que les différences obtenues sont assimilables à du bruit numérique (de l'ordre de 10^{-12} , 10^{-13}). Pour ces deux figures, on a uniquement tracé la composante présentant la plus grande différence entre les résultats des deux simulations. Ces deux résultats nous montrent que la cohérence entre les deux algorithmes est bien vérifiée et nous permet de valider les modèles développés.

Comparaison avec le robot anguille série

Au chapitre 2, nous avons étudié le modèle dynamique d'un robot à structure sérielle. Si la structure est différente de celle étudiée dans ce chapitre, en utilisant la même description géométrique de l'anguille (donc en utilisant les mêmes paramètres dynamiques), on doit alors obtenir les mêmes résultats de simulation. On utilise alors le même générateur de mouvement et le même modèle fluide pour les 2 robots. Les figures (4.9 -a) et (4.9 -b) nous montrent respectivement la vitesse absolue de l'anguille dans le repère terrestre Σ_w et la différence de vitesse entre les simulations des 2 robots :

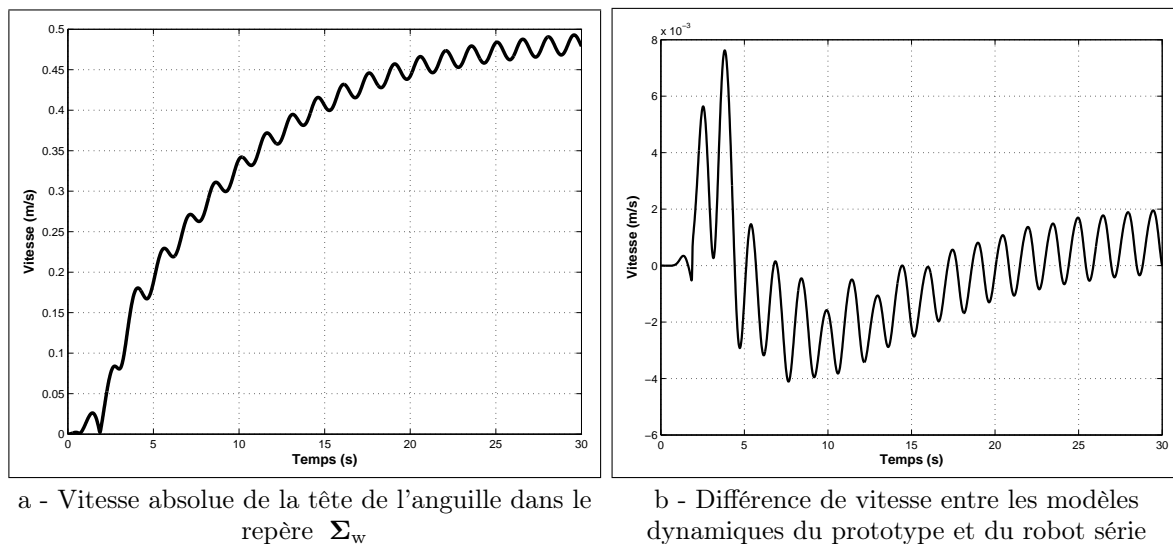
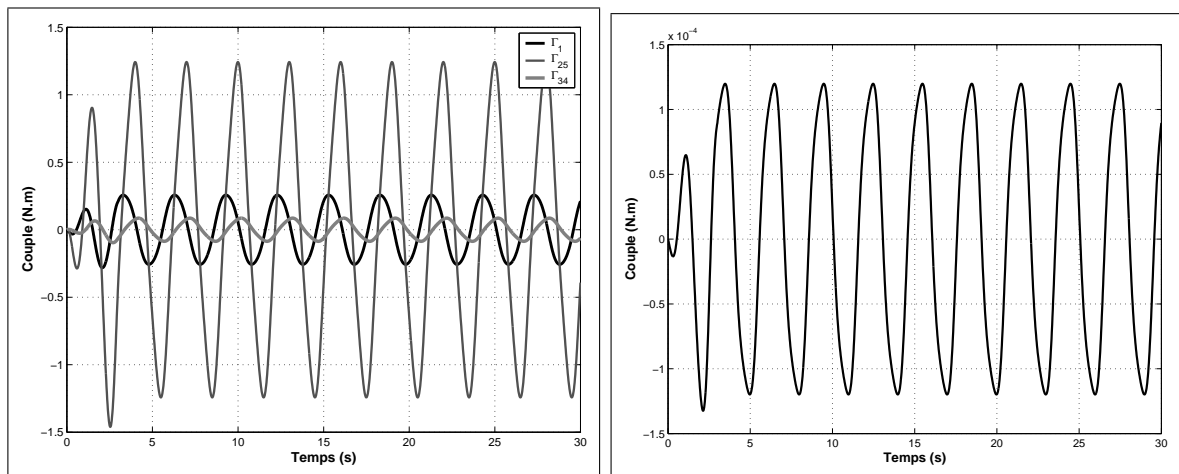


Figure 4.9: – Comparaison de la vitesse entre le robot série et le robot hybride

Les figures (4.9 -a) et (4.9 -b) nous montrent que les résultats obtenus sont semblables, la différence entre les deux modèles étant quasiment nulle. On a donc validé ici les deux premières récurrences de notre algorithme du modèle dynamique inverse.

Pour finir la validation de notre algorithme (dernière récurrence), on compare les couples calculés par les modèles dynamiques des 2 robots lorsque l'anguille est dans le vide. Ainsi, la figure (4.10 -a) montre les couples articulaires des liaisons rotules 1, 9 et 12 (correspondant aux liaisons 1, 25 et 34 pour le robot série), alors que la figure (4.10 -b) montre la liaison rotule pour laquelle on trouve le maximum d'erreurs dans la différence

des couples des deux simulations :



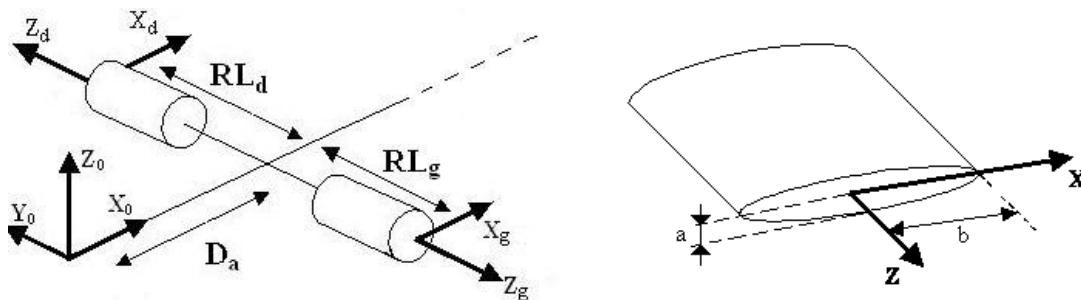
a - Couples articulaires des liaisons rotules 1, 9 et 12 b - Différence de couple pour la liaison rotule N°7

Figure 4.10: – Comparaison des couples entre le robot série et le robot hybride

On remarque que l'erreur maximum entre les couples calculés par les modèles dynamiques inverse du robot série et du robot hybride est de l'ordre de 10^{-4} N.m. Ce dernier test valide une nouvelle fois les algorithmes que nous avons présentés au chapitre 2 et dans celui-ci.

4.7 Modélisation dynamique du robot en ajoutant des ailerons

On a vu dans le chapitre 2 que l'utilisation d'ailerons était nécessaire dès lors qu'on souhaite effectuer des nages 3D. Ces ailerons sont fixés sur la tête de l'anguille et de chaque côté du corps. La figure (4.11) représente la modélisation géométrique adoptée afin de placer les ailerons convenablement :



a - Placement des repères

b - Géométrie des ailerons

Figure 4.11: – Modélisation géométrique avec les ailerons

A partir de cette modélisation géométrique, les matrices de passage permettant d'exprimer les repères Σ_g et Σ_d par rapport au repère de la tête Σ_0 sont les suivantes :

$${}^0\mathbf{T}_g = \begin{bmatrix} \cos(q_{a,g}) & -\sin(q_{a,g}) & 0 & D_a \\ 0 & 0 & -1 & -RL_g \\ \sin(q_{a,g}) & \cos(q_{a,g}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^0\mathbf{T}_d = \begin{bmatrix} \cos(q_{a,d}) & -\sin(q_{a,d}) & 0 & D_a \\ 0 & 0 & 1 & RL_d \\ -\sin(q_{a,d}) & -\cos(q_{a,d}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.108)$$

avec :

- $q_{a,g}$ et $q_{a,d}$ les variables articulaires associées aux ailerons gauche et droit ;
- D_a la distance des ailerons par rapport à la tête ;
- RL_g et RL_d l'écartement des ailerons par rapport à l'axe de la tête.

Pour les simulations on prendra les valeurs numériques telles que :

$$D_a = 0,3 \text{ m} ; RL_g = RL_d = 0,065 \text{ m} ; a = 0,01 \text{ m} ; b = 0,06 \text{ m}.$$

Avec ces données numériques, les paramètres inertiels d'un aileron sont définis par :

- masse : 377 g ;
- moment d'inertie :

$$\mathbf{MS} = [0 \quad 0 \quad 0,0377]^T$$

- matrice d'inertie :

$$\mathbf{J} = \begin{bmatrix} 5,0360.10^{-3} & 0 & 0 \\ 0 & 5,3658.10^{-3} & 0 \\ 0 & 0 & 3,4872.10^{-4} \end{bmatrix}$$

Loi de contrôle pour les ailerons

Pour contrôler le mouvement de roulis, on reprend la loi de contrôle utilisée dans le chapitre 2. Les consignes articulaires affectées aux deux ailerons sont les suivantes :

$$q_{a,g} = q_d - q_a \quad (4.109)$$

$$q_{a,d} = q_d + q_a \quad (4.110)$$

avec :

- $q_d > 0$ pour avoir un mouvement de rotation positif autour de l'axe de la tête \mathbf{x}_0 ;
- $q_a > 0$ pour faire monter l'anguille.

La loi de contrôle adoptée est alors :

$$q_d = -\beta_{\max} \tanh(k_1(\phi - \phi_d) - k_2\omega_{x,0}) \quad (4.111)$$

4.8 Conclusion

Ce chapitre présente les modèles dynamiques inverse et direct du prototype de robot hybride conçu pour le projet robot anguille. Les algorithmes proposés permettent d'obtenir un modèle dynamique 3D qui peut être facilement utilisé en simulation. Les modèles dynamiques sont développés en utilisant le formalisme récursif de Newton-Euler, de telle manière que le modèle inverse permet de calculer les accélérations de la tête et les couples moteurs des articulations, tandis que le modèle direct permet de calculer les accélérations de la tête et les accélérations des plates-formes des modules (puis les accélérations articulaires par le biais de la matrice jacobienne). Nous avons pu également exprimer les modèles géométrique et cinématique inverses des modules qu'ils soient à l'endroit ou à l'envers, en fonction des variables d'état du robot. Finalement, nous avons pu valider les algorithmes en comparant les résultats obtenus par les modèles dynamiques du robot série et du robot hybride et en montrant qu'ils sont identiques.

L'algorithme de calcul du modèle dynamique inverse présenté dans ce chapitre et dont on se sert pour les simulations de nage, possède la caractéristique très intéressante de pouvoir être scindé en deux blocs :

- un premier bloc qui prend en compte l'architecture mécanique du robot et qui permet donc de calculer les vitesses (\mathbf{w}_k) et accélérations ($\dot{\mathbf{w}}_k$) relatives entre la plate-forme et la base en fonction des variables de commande. Il permet également de calculer les couples articulaires $\mathbf{\Gamma}_k$ en fonction des efforts appliqués sur la plate-forme ${}^k\mathbf{f}_k$;
- un second bloc qui permet, grâce à l'algorithme de Newton-Euler, de calculer l'accélération de la tête ${}^0\ddot{\mathbf{V}}_0$ ainsi que le torseur des efforts ${}^k\mathbf{f}_k$ en fonction des vitesses (\mathbf{w}_k) et accélérations ($\dot{\mathbf{w}}_k$) relatives entre chaque module.

L'intérêt de ce découpage est de pouvoir écrire un algorithme de calcul du modèle dynamique ne faisant intervenir (dans ces équations) aucun terme qui dépende de la structure mécanique des modules (comme par exemple \mathbf{J}_k).

Chapitre 5

Simulation de la commande de l'anguille

5.1 Introduction

Ce chapitre est dédié à l'étude de la commande du robot anguille en boucle fermée. Compte tenu du grand nombre de degrés de liberté externes que possède notre robot (position et orientation de la tête, vitesse de l'anguille, nage en avant ou en arrière,...), il doit avoir une certaine autonomie dans la génération de ses mouvements afin que l'opérateur puisse le contrôler. La résolution de ce problème est traitée dans la littérature en introduisant les notions de suivi de trajectoires prédéfinies ou bien de stratégies permettant de se diriger vers un ensemble de points cibles [McIsaac et Ostrowski, 2000, Morgansen *et al.*, 2001, Mason, 2003, Yu *et al.*, 2003, Lapierre et Jouvencel, 2005, Matsuno et Sato, 2005, Alamir *et al.*, 2007]. L'interaction avec l'environnement est également primordial dans le comportement animal. Parmi l'ensemble des capteurs qui permettent de percevoir l'environnement on distinguera deux groupes :

- les capteurs de proximité qui permettent par exemple de détecter les obstacles et donc de naviguer dans des espaces confinés [Sfakiotakis *et al.*, 2006] ;
- les capteurs sensitifs qui permettent de renseigner l'animal sur le milieu dans lequel il évolue (capteur d'eau pour savoir si le serpent est sur le sol ou dans l'eau, capteur de pression pour savoir si l'anguille est en surface ou sur le fond marin,...) [Ijspeert *et al.*, 2007, Sfakiotakis *et al.*, 2007].

Les données provenant de ces différents capteurs influencent la loi de mouvements du corps de l'animal de façon à réagir à la situation courante (tourner pour éviter un obstacle par exemple). L'adaptation à ces différents signaux de stimulation fait que la loi de mouvements internes du corps d'une anguille peut devenir extrêmement complexe. Si les lois horaires analytiques ont été largement utilisées dans la littérature [McIsaac et Ostrowski,

1999, Liu et Hu, 2004, Yu *et al.*, 2005], le contrôle en boucle fermée des robots a poussé les chercheurs à se tourner vers l'étude de générateurs de mouvements bio-inspirés. A partir des travaux d'Ekeberg [Ekeberg, 1993] sur l'étude du rôle et du fonctionnement des différents neurones situés dans la moelle épinière, différents chercheurs ont adapté ces travaux afin d'obtenir des modèles équivalents et simplifiés [Inoue *et al.*, 2004, Righetti *et al.*, 2005]. Ces modèles simplifiés, qu'on appelle CPG (Central Pattern Générateur ou générateur de patron) sont un ensemble d'équations non-linéaires qui convergent vers un cycle limite qu'il est possible de modifier au cours du temps. Un autre avantage des CPGs est leur capacité à se synchroniser sur un signal externe et même d'en apprendre ses caractéristiques.

Pour réaliser les simulations de l'anguille en boucle fermée, nous avons testé un ensemble de méthodes proposées dans la littérature sur le simulateur présenté dans le chapitre 4. Ces différentes méthodes sont organisées de la manière suivante dans ce chapitre : dans le paragraphe 5.2, nous commençons par présenter le générateur de mouvements de [Ijspeert et Crespi, 2007] basé sur un réseau de CPGs répartis le long du corps de l'anguille. Ensuite, le paragraphe 5.3 propose d'utiliser un filtre de Hopf afin d'obtenir l'angle de cap de l'anguille dont on se sert pour la commande.

A partir de ces méthodes, nous présentons et appliquons dans le paragraphe 5.4 la méthodologie de commande de [Alamir *et al.*, 2007] pour le contrôle de l'anguille lorsqu'on souhaite faire du suivi de points dans un espace 3D. Enfin, dans le paragraphe 5.5, nous proposons de combiner le générateur de mouvements à base de CPGs avec des capteurs de distance ajoutés sur la tête de l'anguille afin de faire de l'évitement d'obstacle.

5.2 Commande de l'anguille

5.2.1 Générateur de mouvements

Le problème de la génération de mouvements en boucle ouverte pour le robot anguille a été traité dans les chapitres précédents. Afin d'avoir une approche plus bio-mimétique de la commande du robot, nous avons décidé dans ce chapitre d'utiliser un générateur de mouvements basé sur un réseau de CPGs qui modélise le fonctionnement des neurones d'une anguille biologique. Parmi les différents modèles qui existent dans la littérature, nous avons choisi d'utiliser celui proposé par [Ijspeert et Crespi, 2007]. Le choix de ce type de générateur de mouvements est motivé par le caractère adaptatif du comportement des CPGs. Les équations qui régissent un CPG possèdent le double avantage de filtrer et converger de manière automatique vers les nouvelles valeurs de consigne.

Pour rappel, un CPG est un ensemble d'équations non-linéaires qui permettent de générer un signal sinusoïdal. Pour générer l'onde progressive qui se propage le long du corps de

l'anguille, les auteurs proposent d'utiliser un réseau de CPGs selon un schéma représenté sur la figure (5.1) :

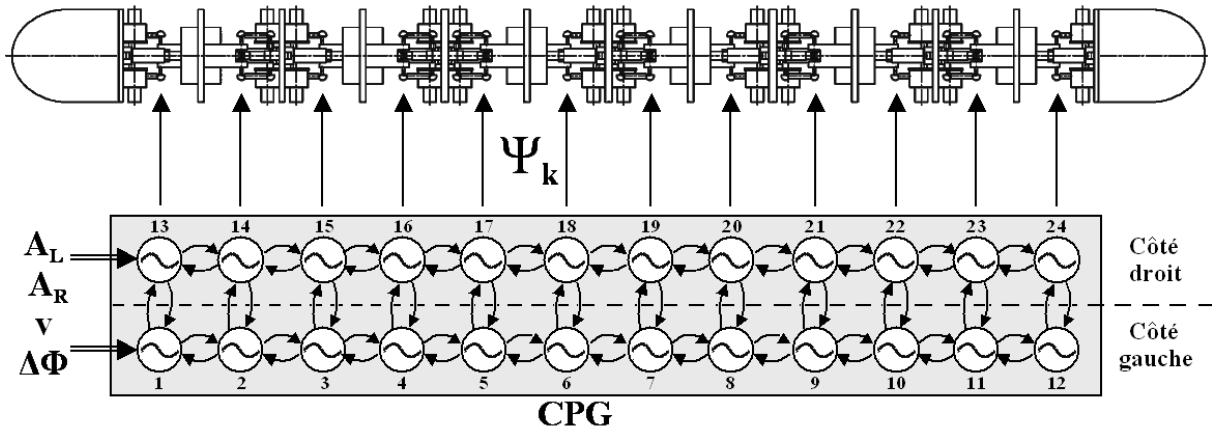


Figure 5.1: – Générateur de propulsion à base de CPG

Le générateur de propulsion proposé par [Ijspeert et Crespi, 2007] possède 2 chaînes de CPGs (avec dans notre cas 12 CPGs par chaîne pour les 12 articulations du robot) pour modéliser les muscles des côtés droit et gauche du corps de l'anguille. Les couplages bilatéraux entre les CPGs permettent de créer d'une part le déphasage entre les corps afin de générer l'onde de propulsion et, d'autre part la synchronisation entre les muscles des côtés droit et gauche. Le comportement global de ce générateur est paramétré par 4 variables :

- A_L et A_R qui représentent l'amplitude des signaux sinusoïdaux qui sont produits par les chaînes de CPG des côtés gauche et droit ;
- v qui représente la fréquence du signal sinusoïdal (donc de l'onde qui traverse le corps de l'anguille) ;
- $\Delta\phi$ qui représente le déphasage des signaux sinusoïdaux entre chaque articulation du robot.

Ainsi, on remarque qu'en agissant sur les valeurs A_L et A_R , on pourra agir sur la courbure du robot afin de le faire tourner.

Pour un CPG i , les équations mathématiques permettant de générer les signaux sinusoïdaux sont les suivantes [Ijspeert *et al.*, 2007] :

$$\dot{p}_i = 2\pi v_i + \sum_j w_{ij} \sin(p_j - p_i - \Delta p_{ij}) \quad (5.1)$$

$$\ddot{r}_i = a_i \left(\frac{a_i}{4} (R_i - r_i) - \dot{r}_i \right) \quad (5.2)$$

$$x_i = r_i (1 + \cos(p_i)) \quad (5.3)$$

avec pour paramètres :

- p_i qui représente la phase du signal sinusoïdal ;
- r_i qui représente l'amplitude courante du signal ;
- v_i qui représente la fréquence du signal ;
- R_i qui représente l'amplitude souhaitée des oscillations ;
- a_i une constante positive qui paramètre la rapidité de convergence de l'oscillateur.

Le couplage entre les différents CPGs se fait par le biais de w_{ij} qui représente le poids de couplage et Δp_{ij} qui représente le retard de phase souhaité entre les deux oscillateurs couplés i et j . Enfin, x_i représente le signal de sortie.

Fonctionnement d'un oscillateur seul :

Avant de commencer l'étude du générateur de mouvements complet, nous analysons le comportement d'un CPG seul soumis aux signaux excitateurs de fréquence (v) et d'amplitude (R) (voir figure 5.2)



Figure 5.2: – Etude du comportement d'un CPG seul

Ainsi, la figure (5.3) montre la sortie de ce CPG. On observe alors son comportement en fonction des consignes des entrées v et R :

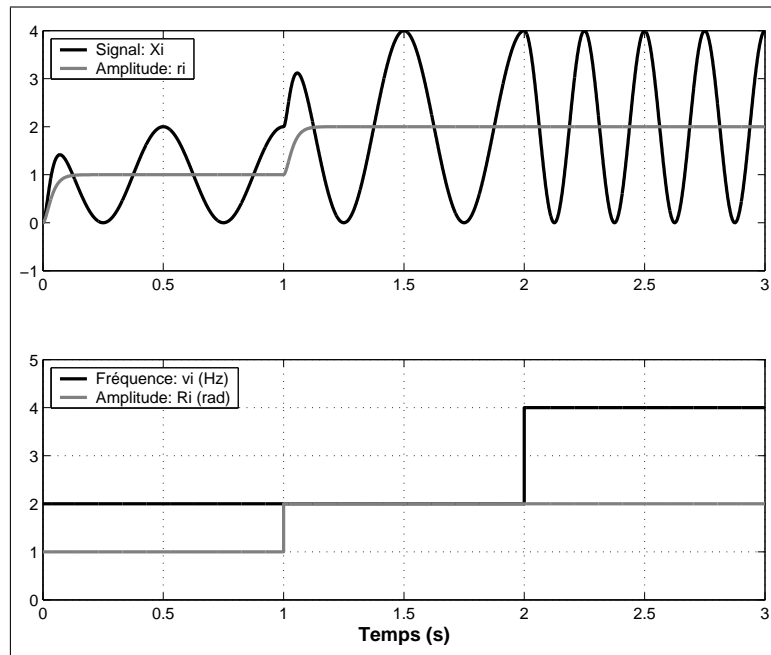


Figure 5.3: – Simulation du CPG pour différentes valeurs de v et R et $a_i=100$

On remarque sur la figure (5.3) qu'en imposant un échelon d'amplitude, le signal r_i (correspondant à l'amplitude courante du CPG) converge progressivement vers la nouvelle valeur de consigne. A partir de l'équation (5.1) et en en tenant compte que le CPG est seul (les termes de la somme disparaissent), on s'aperçoit que la variation de phase \dot{p}_i est réduite à une simple constante :

$$\dot{p}_i = 2\pi v_i$$

Par conséquent, lors d'un changement de consigne en fréquence, celle-ci sera immédiatement appliquée sans phase de transition.

Fonctionnement d'un oscillateur couplé :

Afin d'analyser le comportement de plusieurs CPGs couplés entre eux, nous simulons le comportement de 2 CPGs soumis aux signaux excitateurs de fréquence (v), d'amplitude (R_1 et R_2) et de déphasage $\Delta\phi$ (voir figure 5.4) :

Contrairement au cas du CPG seul, on introduit cette fois-ci un terme de déphasage $\Delta\phi$ qui traduit la différence de phase souhaitée entre les signaux x_1 et x_2 des CPGs droit et gauche. On prendra les coefficients suivants :

- $v_i = v = 1$ Hz ;
- $R_1 = R_2 = 1$;
- $a_i = 100$;

- $w_{ij} = 4$;
- $\Delta p_{ij} = e_i \Delta \phi$, avec $e_i = -1$ pour $i=1$ et $e_i = 1$ pour $i=2$.

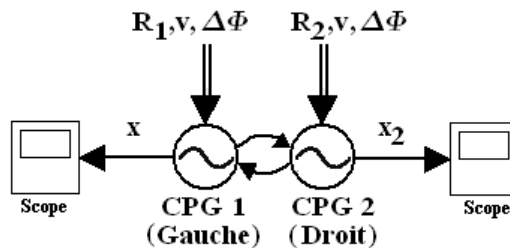


Figure 5.4: – Etude du comportement de deux CPG Couplés

On note qu'entre 2 CPGs couplés, les déphasages doivent être de signes opposés. Dans notre cas, le CPG gauche est en retard de phase par rapport au CPG droit (inversement le CPG droit est en avance de phase par rapport à celui de gauche). La figure (5.5) montre le comportement des 2 CPGs en fonction du déphasage $\Delta \phi$. On y trace également le déphasage qui existe à chaque instant entre les deux CPG en utilisant la relation :

$$\Delta p(t) = p_1 - p_2$$

Pour cette simulation, toutes les conditions initiales sont nulles :

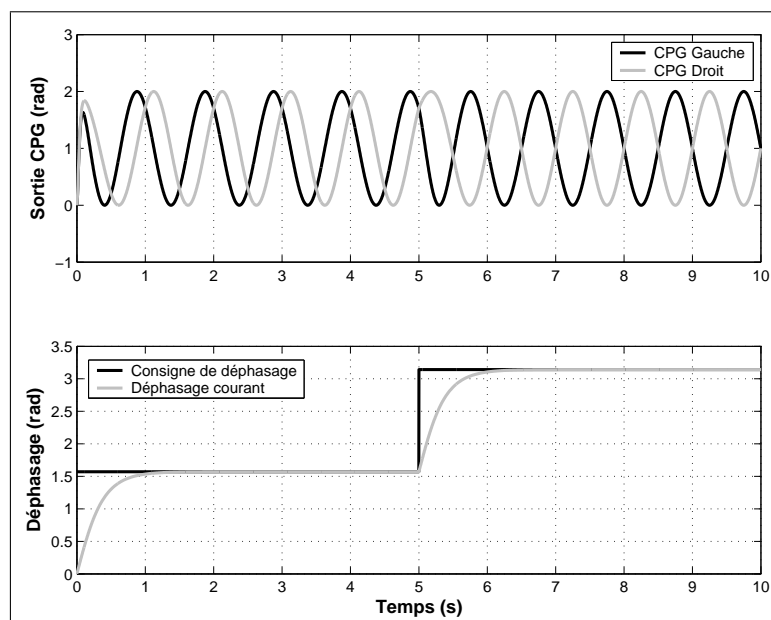


Figure 5.5: – Simulation de deux CPGs pour différentes valeurs de déphasage $\Delta \phi$

La figure (5.5) montre bien que pour des consignes en échelon de $\Delta \phi$, le déphasage entre les deux oscillateurs convergent progressivement vers la valeur désirée de manière continue.

Il est à noter que la rapidité de convergence du déphasage peut être réglée en agissant sur la valeur du poids de couplage w_{ij} .

Simulation du générateur complet :

La consigne d'angle en lacet ψ_k qui doit être appliquée pour chaque module est donnée par la différence des signaux entre les oscillateurs de la chaîne gauche et droite. Si on numérote $i=[1,...,N]$ ($N=12$) les oscillateurs de la chaîne gauche et $i=[N+1,...,2N]$ les oscillateurs de la chaîne droite, les angles ψ_k sont calculés par la relation :

$$\psi_k = x_i - x_{N+i} \quad (5.4)$$

Les relations qui permettent de relier les 4 paramètres de commande ($A_L, A_R, \omega, \Delta\phi$) et les paramètres de l'oscillateur sont :

- $R_i = A_L$ pour les oscillateurs du côté gauche et $R_i = A_R$ pour les oscillateurs du côté droit ;
- $\Delta p_{ij} = \Delta\phi$ pour les connexions descendantes entre les oscillateurs ($i \rightarrow i+1$) et $\Delta p_{ij} = -\Delta\phi$ pour les connexions montantes entre les oscillateurs ($i+1 \rightarrow i$) ;
- $\Delta p_{ij} = \pi$ pour les connexions entre les oscillateurs ($i \leftrightarrow i+N$) des côtés droit et gauche ;
- $v_i = v$ qui représente la fréquence souhaitée de l'ondulation.

Avec ces paramètres, il a été démontré dans [Ijspeert et Crespi, 2007] que l'angle ψ_k converge vers la valeur suivante :

$$\psi_k^\infty(t) = A_L - A_R + (A_L + A_R) \cos(2\pi vt + k\Delta\phi + \phi_0) \quad (5.5)$$

Pour un robot possédant 7 articulations, la figure (5.6) montre les consignes $\psi_k(t)$ que les auteurs ont obtenues pour différentes valeurs de A_L, A_R , de fréquence (v) et de phase ($\Delta\phi$) :

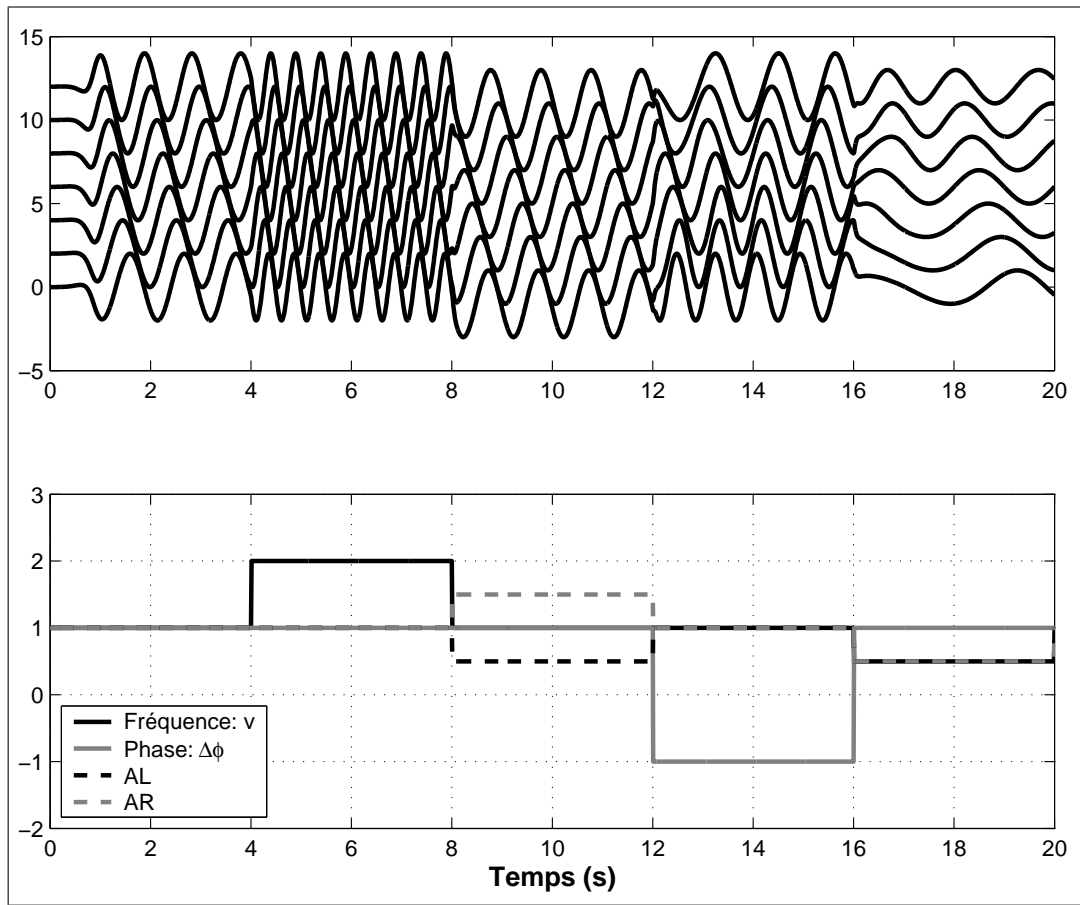


Figure 5.6: – Simulation d'un réseau de CPG pour différentes valeurs des paramètres de contrôle

La figure (5.6) permet d'analyser les résultats pour 5 jeux de paramètres :

- Pour $t \in [0; 4s]$, $A_L = A_R = 1$, $v=1$ Hz et $\Delta\phi = 1$ rd : on peut remarquer que sur cette période les signaux sont décalés dans le temps, ce qui traduit l'onde de propulsion pour le robot ;
- Pour $t \in [4; 8s]$, on modifie uniquement la fréquence à $v=2$ Hz : le décalage des signaux reste le même que précédemment mais on observe que les signaux possèdent une fréquence d'oscillation deux fois plus élevée comme prévu ;
- Pour $t \in [8; 12s]$, la fréquence redevient $v=1$ Hz. On modifie A_L et A_R pour qu'ils deviennent respectivement égaux à 0,5 et 1,5 : le signal redevient identique à celui pour $t \in [0; 4s]$. Seulement, on observe que les courbes présentent un décalage (offset). Dans ce cas, le robot va se courber et effectuer un virage ;
- Pour $t \in [12; 16s]$, les paramètres reprennent leurs valeurs initiales sauf le déphasage $\Delta\phi$ que l'on modifie et qui prend comme valeur -1 rd : on observe alors un changement dans le décalage des signaux. Le signal qui était le plus en retard devient celui qui est le plus en avance. Dans ce cas de figure le robot ne va plus avancer mais reculer ;
- Pour $t \in [16; 20s]$, $\Delta\phi$ reprend sa valeur initiale et on modifie A_L et A_R pour qu'ils

prennent la valeur 0,5 rd : on remarque alors que l'amplitude des signaux a diminué de moitié par rapport aux cas précédents.

Cette simulation nous a montré que les coefficients A_L et A_R permettent de contrôler l'amplitude des oscillations ainsi qu'un décalage (offset) ajouté aux signaux. Pour trouver une meilleure relation entre le couple (amplitude,offset) et le couple (A_L, A_R) , on réécrit l'équation (5.5) sous la forme :

$$\psi_k^\infty(t) = O_{\text{ffs}} + A \cos(2\pi vt + k\Delta\phi + \phi_0) \quad (5.6)$$

On peut alors en déduire les valeurs des consignes A_L et A_R :

$$A_L = \frac{A}{2} + \frac{O_{\text{ffs}}}{2} \quad (5.7)$$

$$A_R = \frac{A}{2} - \frac{O_{\text{ffs}}}{2} \quad (5.8)$$

Pour les deux autres angles $\phi_k(t)$ et $\theta_k(t)$ de la liaison rotule de chaque module k , la littérature ne propose pas de méthodes à base de CPGs. Ces articulations n'ayant pas besoin de générer de signal sinusoïdal, mais simplement de converger vers une valeur de consigne, nous proposons d'utiliser les équations suivantes pour $k=1, \dots, N$ ($N=12$) :

$$\dot{\phi}_k(t) = \frac{\phi_{k-1}(t) - \phi_k(t)}{\tau} \quad (5.9)$$

$$\phi_0(t) = \phi_c(t) \quad (5.10)$$

$$\dot{\theta}_k(t) = \frac{\theta_{k-1}(t) - \theta_k(t)}{\tau} \quad (5.11)$$

$$\theta_0(t) = \theta_c(t) \quad (5.12)$$

Avec $\phi_c(t)$ et $\theta_c(t)$ représentant les consignes d'angles de tangage et de roulis appliquées sur la première articulation rotule. Ces consignes sont spécifiées par l'opérateur ou par une loi de commande. Les équations (5.9) et (5.11) permettent ensuite d'obtenir un effet de propagation des consignes d'une articulation k à une articulation $k+1$. Enfin, $\dot{\phi}_k$ et $\dot{\theta}_k$ représentent les vitesses angulaires de ϕ_k et θ_k .

On remarque que ces équations sont l'expression d'une fonction de transfert du premier ordre dont l'entrée est l'angle de l'articulation précédente. En effet, pour $k=1$, on a pour l'angle de tangage :

$$\phi_1(t) = \frac{1}{1 + s\tau} \phi_c(t) \quad (5.13)$$

avec τ , la constante de temps permettant de régler la vitesse de convergence de la fonction de transfert.

5.2.2 Simulation du prototype

Pour les simulations du prototype, on utilise le modèle dynamique présenté dans le chapitre 4, en remplaçant la loi de mouvements analytique par le générateur de mouvements à base de CPGs. Si ces deux générateurs permettent d'obtenir les mêmes résultats en régime permanent, l'utilisation des CPGs possède plusieurs avantages :

- l'adaptation à de nouvelles consignes de commande est plus simple puisqu'ils s'adaptent et qu'ils appliquent un filtrage à ces entrées ;
- en connectant des capteurs sur les CPGs, chaque vertèbre peut modifier de manière indépendante son ondulation (capteurs de proximité par exemple) ;
- le robot peut être assez facilement reconfiguré dans le cas d'un module en panne.

Préliminaires

Même si nous utilisons les CPGs pour les simulations, on souhaite pouvoir paramétrer le réseau de neurones avec les mêmes paramètres de base que pour la loi analytique. Pour rappel, cette loi de mouvements que nous avons utilisée précédemment est de la forme :

$$\psi(s, t) = f(t) \cdot A \cdot e^{\alpha s} \cdot \sin \left[2\pi \left(\frac{s}{\lambda} - \frac{t}{T} \right) \right] \quad (5.14)$$

Pour laquelle :

- A : amplitude du mouvement ;
- α : réglage de l'exponentielle : augmentation de l'amplitude de la tête vers la queue ;
- T : période de l'oscillation ;
- λ : longueur d'onde de l'oscillation ;
- s : abscisse curviligne le long de la fibre neutre de l'anguille ;
- $f(t)$: fonction de lissage du démarrage et de l'arrêt ;

La fonction $f(t)$ est une fonction polynomiale du 5ème degré :

$$\begin{aligned} f(t) &= a_0 \cdot t^5 + a_1 \cdot t^4 + a_2 \cdot t^3 + a_3 \cdot t^2 + a_4 \cdot t + a_5 & 0 \leq t \leq t_f \\ f(t) &= 1 & t > t_f \end{aligned} \quad (5.15)$$

Les coefficients a_i de ce polynôme satisfont les conditions aux limites suivantes :

$$f(0) = 0, f(t_f) = 1, \dot{f}(0) = \ddot{f}(0) = 0, \dot{f}(t_f) = \ddot{f}(t_f) = 0$$

où t_f représente le temps de fin de $f(t)$.

Pour pouvoir faire des virages à l'anguille nous avons ajouté un terme d'"offset" O_{ffs} sur chacune des articulations.

A partir de cette loi, nous allons adapter ces paramètres pour qu'ils soient compatibles avec le générateur à base de CPGs.

Calcul de l'amplitude du mouvement des articulations :

On peut voir sur l'équation (5.14) que l'amplitude des oscillations articulaires augmente de la tête vers la queue de l'anguille. Or, jusqu'à présent pour le générateur à base de CPGs, nous avons appliqué un signal neuronal identique (A_L ou A_R suivant la chaîne considérée) comme consigne d'amplitude pour les CPGs d'une chaîne. Dans ce cas, l'amplitude des oscillations articulaires ne peut pas être augmentée le long du corps. En réalité, cette augmentation d'amplitude provient de l'arrangement des CPGs : pour chaque CPG dans une chaîne, le signal neuronal provenant du CPG précédent est amplifié par la connexion synaptique [Ekeberg, 1993]. Ainsi, l'amplitude R_i de chaque CPG est calculée de la manière suivante pour $i=1, \dots, N$ ($N=12$) :

$$R_i = A_{g,i} \quad , \quad R_{N+i} = A_{d,i} \quad (5.16)$$

avec :

$$A_{g,1} = A_L \quad , \quad A_{d,1} = A_R \quad (5.17)$$

$$A_{g,i} = A_{g,i-1}e^{\alpha Lc} \quad , \quad A_{d,i} = A_{d,i-1}e^{\alpha Lc} \quad (5.18)$$

où :

- R_i représente la consigne d'amplitude du CPG i ;
- α est une constante qui définit l'augmentation d'amplitude entre chaque CPG d'une chaîne ;
- Lc qui représente la longueur de chaque corps (dans notre cas $Lc=0,14$ cm).

On définit également :

$$A_L = \frac{f(t).A_{LR}}{2} + \frac{O_{ffs}}{2} \quad (5.19)$$

$$A_R = \frac{f(t).A_{LR}}{2} - \frac{O_{ffs}}{2} \quad (5.20)$$

où $f(t)$ est une fonction de démarrage dont l'expression est celle donnée par l'équation (5.15) et A_{LR} la consigne d'amplitude des oscillations pour la première paire de CPGs.

Calcul du déphasage entre chaque corps :

Dans l'équation (5.14), le déphasage après chaque corps est paramétré par l'abscisse curviligne (s) où se situe l'articulation et par la longueur d'onde (λ) de l'oscillation. La longueur d'onde d'un signal correspondant à un déphasage complet de 2π , on peut alors en déduire le déphasage qui existe entre chaque articulation du robot :

$$\Delta\phi = \frac{2\pi Lc}{\lambda} \quad (5.21)$$

Avec $\Delta\phi$ le déphasage entre deux CPGs consécutifs d'une même chaîne.

Simulateur pour la nage du prototype

La simulation du prototype est réalisée sous Matlab-Simulink dont le schéma simulink est représenté sur la figure (5.7). Le comportement dynamique du robot pendant la nage est déterminé en utilisant les équations du modèle dynamique du prototype (modélisation hybride du robot) présentées dans le chapitre 4.

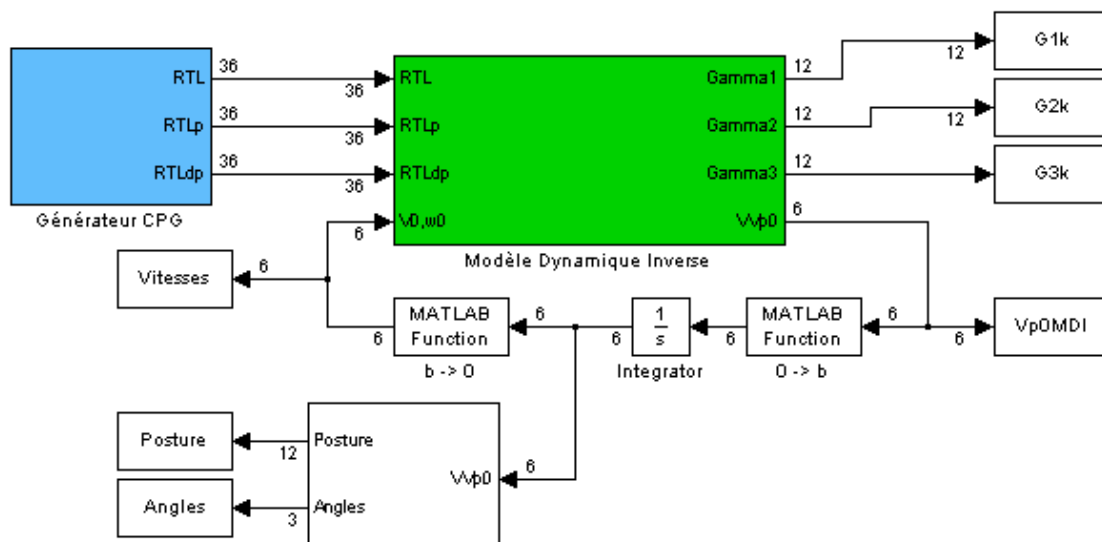


Figure 5.7: – Simulateur de l'anguille avec un générateur de mouvements à base de CPG

Le détail du bloc "GénérateurCPG" qui permet de calculer les consignes articulaires est représenté sur la figure (5.8) :

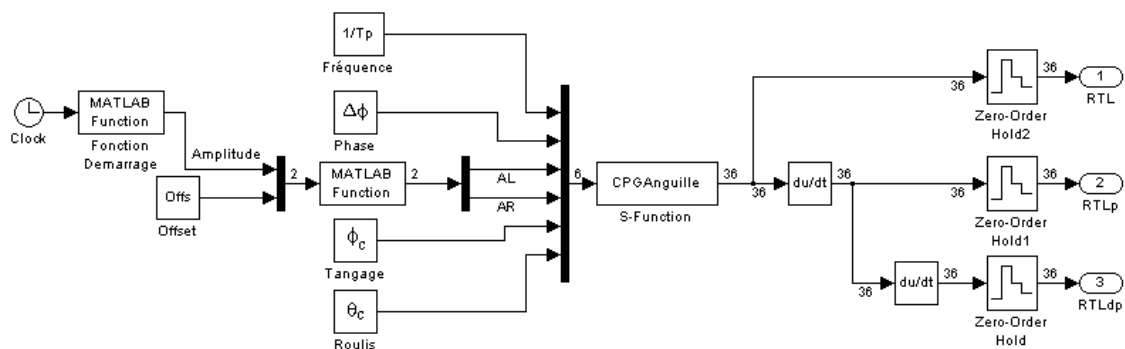


Figure 5.8: – Schéma simulink du générateur de mouvements pour le calcul des consignes de Roulis-Tangage-Lacet

Le générateur de mouvements que nous avons étudié (bloc CPGAnguille) permet de calculer les positions articulaires à appliquer au robot. Or, les algorithmes du modèle dynamique

ayant besoin des vitesses et accélérations articulaires, il est alors nécessaire de dériver les consignes de positions pour obtenir les vitesses et accélérations articulaires.

Pour vérifier le fonctionnement du générateur complet, on effectue une simulation avec les paramètres suivants :

$$A_{LR}=0,15 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; \Delta\phi = \frac{2\pi Lc}{\lambda} ; Lc=0,14 \text{ m} ; \lambda=1,5 \text{ m} ; v=0.33 \text{ Hz} ; t_f=4 \text{ s} ; \\ O_{fs}=0 ; \phi_c = 0 ; \theta_c = 0$$

La figure (5.9) montre les angles de lacet générés par les CPGs des liaisons rotules 1, 4, 7 et 10 :

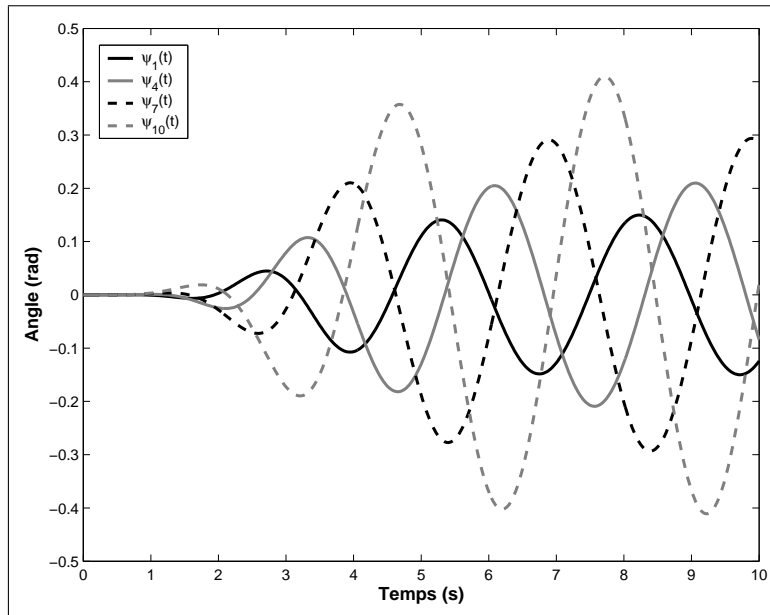


Figure 5.9: – Loi horaire des angles de lacet pour différentes liaisons rotules

En analysant la figure (5.9), on retrouve les caractéristiques souhaitées d'un déphasage constant entre les différentes articulations ainsi qu'une augmentation de leur amplitude lorsqu'on se dirige vers la queue de l'anguille. On peut également observer que la fonction $f(t)$ permet pendant les premières secondes de la simulation d'avoir une augmentation progressive de l'amplitude des signaux, ceci afin d'avoir un démarrage de l'anguille en douceur.

Pour tester le comportement des angles de tangage $\phi_k(t)$ de chaque rotule, on applique sur la première articulation une consigne de la forme :

$$\phi_c = f(t)O_{fs2}$$

avec $O_{fs2}=0,08 \text{ rad}$. La vitesse de convergence de la fonction de transfert pour chaque rotule (équation 5.9) est réglée par $\tau=0,2$.

La loi horaire obtenue est tracée sur la figure (5.10) pour les angles de tangage des liaisons rotules 1, 4, 7 et 10 :

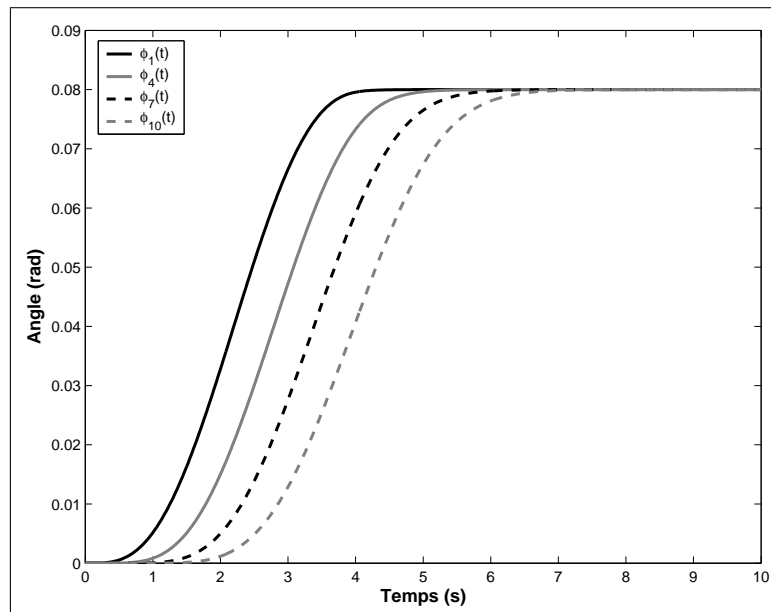


Figure 5.10: – Loi horaire des angles de tangage pour différentes liaisons rotules

Cette figure montre que la fonction $f(t)$ permet d'appliquer l'offset souhaité avec une loi d'évolution lisse. On remarque également que les courbes des différentes articulations sont décalées dans le temps ce qui permet d'obtenir un effet de propagation de la consigne le long du corps de l'anguille.

Pour les angles de roulis θ_k , l'équation d'évolution (5.11) étant la même que pour les angles de tangage (équation 5.9), alors le comportement sera similaire.

5.3 Méthode de mesure du cap de l'anguille

5.3.1 Présentation du problème

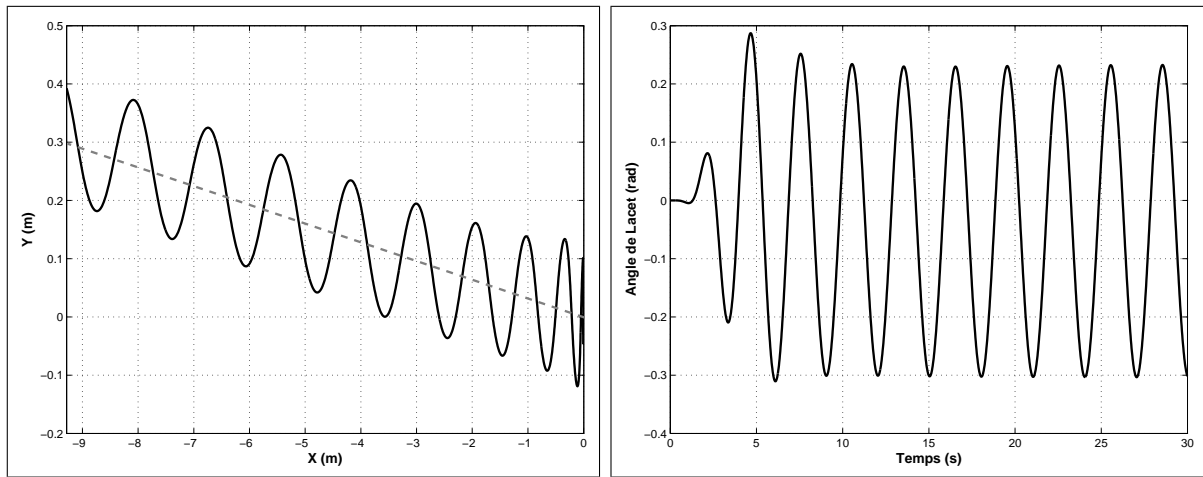
Jusqu'ici, toutes les simulations du robot anguille ont été réalisées en boucle ouverte. Pour une commande en boucle fermée nous supposons disponibles par des capteurs appropriés les informations suivantes :

- position de l'anguille dans un repère fixe pour de la poursuite de cible ;
- vitesse de l'anguille si l'on souhaite réaliser un asservissement en vitesse ;
- orientation de la tête obtenue par une centrale inertielle ;
- une mesure de la position de la cible à atteindre par rapport à la tête de l'anguille.

Cependant, certaines de ces données ne sont pas directement exploitables pour la commande. Prenons le cas de la nage plane pour laquelle on souhaite estimer le cap de l'anguille à tout instant. Pour analyser les informations en notre possession, on réalise une simulation avec les paramètres suivants :

$$A_{LR}=0,15 \text{ rad}; \alpha=0,8 \text{ m}^{-1}; \Delta\phi = \frac{2\pi Lc}{\lambda}; Lc=0,14 \text{ m}; \lambda=1,5 \text{ m}; v=0,33 \text{ Hz}; t_f=4 \text{ s}; \\ O_{ffs}=0; \phi_c = 0; \theta_c = 0$$

Les figures (5.11 -a) et (5.11 -b) représentent la position de la tête dans le plan x-y du repère fixe ainsi que l'angle de lacet de la tête ψ_0 dans ce même repère obtenus par la simulation :



a - Position de l'origine de la tête dans le plan x-y du repère fixe

b - Angle de lacet ψ_0 de la tête

Figure 5.11: – Données issues des différents capteurs de l'anguille pour estimer le cap

La figure (5.11 -a) montre qu'en boucle ouverte, l'anguille nage selon un angle de cap déterminé ψ_c , différent de l'axe x (axe en pointillé gris), qu'il faut pouvoir estimer. La mesure de cet angle de cap ψ_c peut être faite à partir de l'angle de lacet de la tête. Sur la figure (5.11 -b), cet angle correspond à l'offset présent dans le signal sinusoïdal.

Pour estimer l'offset, donc le cap de l'anguille, une première méthode utilisée par [Alamir *et al.*, 2007] consiste à calculer la moyenne temporelle sur une période de ce signal.

Pour rester dans le cadre bio-mimétique, nous proposons une autre méthode dont la motivation provient des caractéristiques du signal de l'angle de lacet :

- le signal est sinusoïdal ;
- le signal possède une fréquence égale à celle de l'onde du générateur de mouvements.

Le calcul de la valeur moyenne du signal (donc du cap de l'anguille) peut se schématiser sous la forme d'un filtre :



Figure 5.12: – Mise en place d'un filtre pour le calcul de la moyenne

où $f(t)$ représente le signal d'entrée et $\tilde{f}(t)$ la valeur moyenne de celui-ci. Pour caractériser ce filtre, nous utilisons un oscillateur de Hopf avec apprentissage afin de reconstruire le signal $f(t)$ et d'en déduire les caractéristiques qui nous intéressent.

5.3.2 Oscillateur de Hopf

Généralités

L'oscillateur de Hopf, comme le CPG étudié précédemment, est un ensemble d'équations non-linéaires ayant pour but de générer un signal sinusoïdal à partir des consignes de fréquence et d'amplitude. A l'origine, cet oscillateur a été utilisé pour modéliser le comportement des réseaux de neurones des poissons. La dynamique de l'oscillateur de Hopf est représentée par les équations ci-dessous :

$$\dot{x} = (\mu - r^2)x - \omega y \quad (5.22)$$

$$\dot{y} = (\mu - r^2)y + \omega x \quad (5.23)$$

avec :

- $r = \sqrt{x^2 + y^2}$, l'amplitude à l'instant t de l'oscillation ;
- $\sqrt{\mu}$, la consigne d'amplitude du signal ;
- ω , le terme de réglage en pulsation de l'oscillateur.

Ce système d'équations permet d'obtenir un système oscillant de pulsation ω (voir figure 5.13-b). Si on trace la réponse de l'oscillateur dans le système de coordonnées (x-y), on observe sur la figure (5.13-a) que celui-ci possède un cycle limite asymptotiquement stable de rayon $\sqrt{\mu}$, quelles que soient les conditions initiales (sauf pour $x(0)=y(0)=0$) :

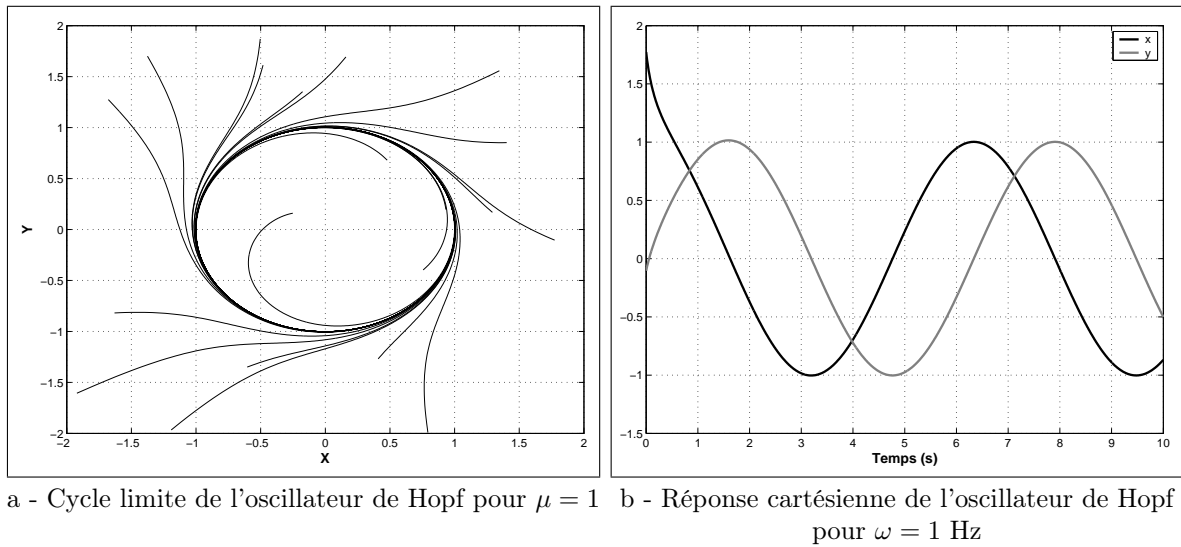


Figure 5.13: – Comportement d'un oscillateur de Hopf

Sur la base de cet oscillateur, Righetti et al. [Righetti *et al.*, 2005] ont proposé de le modifier de telle sorte qu'il soit capable, en plus de ses caractéristiques de base, d'apprendre la fréquence et l'amplitude d'un signal d'apprentissage. Pour ce faire, le système d'équations devient :

$$\dot{x} = (\mu - r^2)x - \omega y + \epsilon F(t) \quad (5.24)$$

$$\dot{y} = (\mu - r^2)y + \omega x \quad (5.25)$$

$$\dot{\omega} = -\epsilon F(t) \frac{y}{r} \quad (5.26)$$

$$\dot{\alpha} = \eta x F(t) \quad (5.27)$$

$$F(t) = P_{teach}(t) - \alpha x \quad (5.28)$$

avec :

- $P_{teach}(t)$ représente le signal d'entrée à apprendre ;
- ϵ et η sont des coefficients permettant de régler la vitesse de convergence ;
- α le terme d'apprentissage de l'amplitude.

En utilisant ces équations, la fréquence ω de cet oscillateur converge vers la fréquence du signal sinusoïdal à apprendre. Pour analyser cette propriété, nous observons le comportement de l'oscillateur pour le signal d'apprentissage suivant :

$$P_{teach} = 0.5 \sin(30t) \quad (5.29)$$

On trace alors sur la figure (5.14) la réponse de l'oscillateur pour plusieurs conditions initiales $\omega(0)$ de ω :

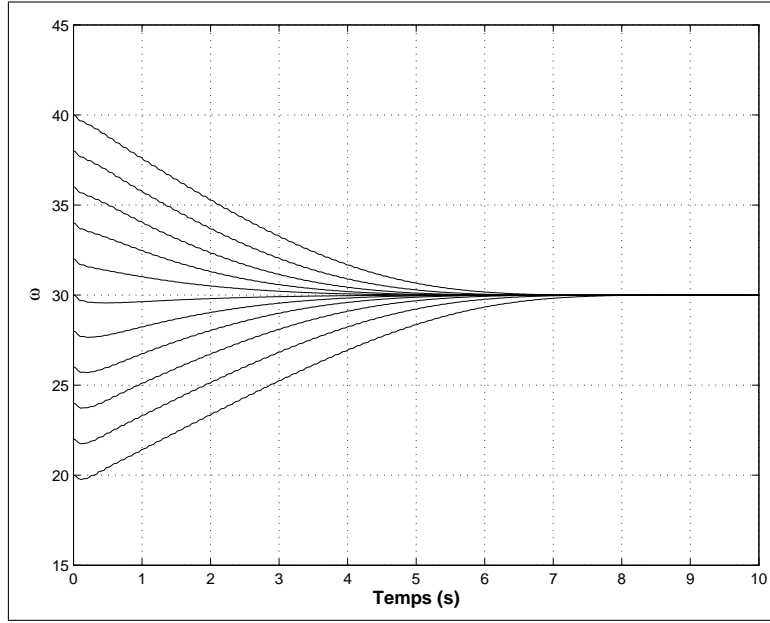


Figure 5.14: – Evolution de ω pour différentes conditions initiales $\omega(0)$ pour la fonction d'apprentissage et $(\mu = 1, \epsilon = 10, \eta = 8)$ comme paramètres

On remarque bien que, quelle que soit la condition initiale $\omega(0)$ choisie, l'oscillateur converge vers la pulsation du signal P_{teach} (ici 30 rd/s).

Oscillateur proposé

L'oscillateur de Hopf que l'on vient d'étudier, nous a permis d'apprendre la fréquence et l'amplitude d'un signal mais il ne permet pas de connaître les éventuels offset qui ont été ajoutés. Dans le cas qui nous intéresse, nous connaissons la fréquence du signal et nous souhaitons estimer son amplitude ainsi que son offset. Pour cela, en se basant sur la méthode d'apprentissage de la fréquence que l'on vient de voir, nous proposons l'oscillateur suivant :

$$\dot{x} = (\mu - r^2)x - \omega y + \epsilon F(t) \quad (5.30)$$

$$\dot{y} = (\mu - r^2)y + \omega x \quad (5.31)$$

$$\dot{\alpha} = \eta F(t) \frac{x}{r} \quad (5.32)$$

$$\dot{\delta} = \lambda \left[\left(P_{teach}(t) - \frac{\alpha x}{r} \right) - \delta \right] \quad (5.33)$$

$$F(t) = P_{teach}(t) - \left(\frac{\alpha x}{r} + \delta \right) \quad (5.34)$$

avec comme nouveaux paramètres :

– δ le paramètre d'apprentissage de l'offset ;

- λ un paramètre de réglage pour la vitesse de convergence de l'apprentissage de l'offset.

Avec cet ensemble d'équations, la valeur moyenne du signal d'apprentissage $\tilde{P}_{\text{teach}}(t)$ est obtenue en prenant comme sortie du filtre la variable δ . Pour tester le filtre, on prend comme signal d'apprentissage le signal de l'angle de lacet ψ_0 obtenu lors d'une simulation de nage (figure 5.11 -b). La valeur filtrée $\tilde{\psi}_0$ de ce signal nous permet d'estimer l'angle de cap courant de l'anguille ψ_c :

$$\psi_c = \tilde{\psi}_0 \quad (5.35)$$

Le réglage que nous avons choisi pour l'oscillateur est le suivant :

$$\mu = 1; \eta = 25; \epsilon = 10; \lambda = 2$$

On choisit par ailleurs $\omega = \frac{2\pi}{T}$ avec $T = 3$ s correspondant à la période de propagation de l'onde lors de la simulation.

L'apprentissage du signal de l'angle de lacet ψ_0 avec l'oscillateur donne le résultat présenté sur la figure (5.15) :

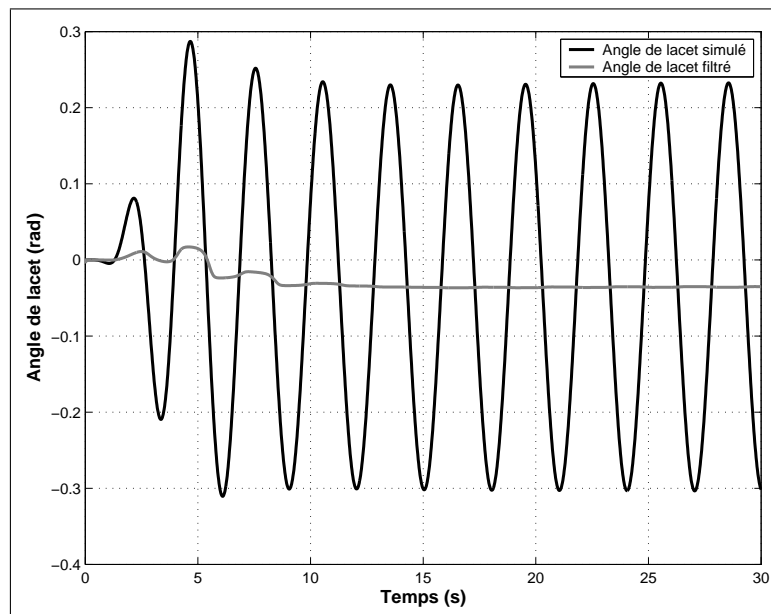
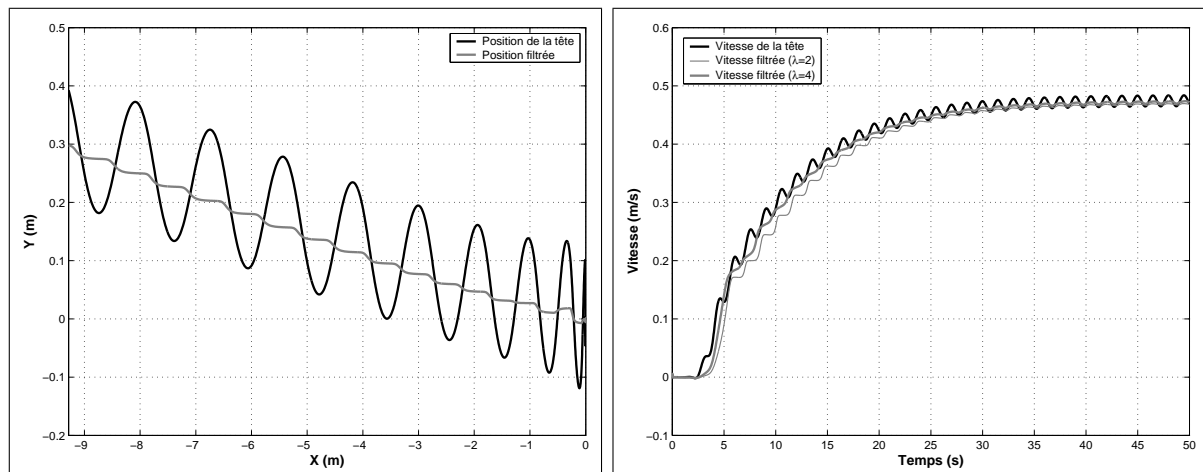


Figure 5.15: – Estimation de l'offset ψ_f du signal à apprendre

Le résultat obtenu sur la figure (5.15) montre que l'offset du signal sinusoïdal est correctement identifié.

On peut également appliquer ce filtre, pour estimer les valeurs moyennes de la position de la tête de l'anguille, ou de sa vitesse :



a - Application de l'oscillateur sur la position de la tête dans le pla x-y du repère fixe b - Application de l'oscillateur sur la vitesse de l'anguille pour différentes valeurs de λ

Figure 5.16: – Application de l'oscillateur proposé pour divers signaux

La figure (5.16 -a) montre que le filtrage de la position de la tête permet d'obtenir une estimation de la trajectoire très proche de la ligne tracée sur la figure (5.11 -a) (en pointillé). La figure (5.16 -b) montre l'importance du réglage du coefficient λ traduisant la vitesse de convergence de l'estimation de l'offset. On remarque que pour $\lambda = 2$, la vitesse de convergence est trop faible, ce qui induit une estimation pessimiste de la vitesse réelle sur le début de la courbe. Lorsqu'on augmente la rapidité de convergence ($\lambda = 4$), le résultat obtenu est meilleur et donne une bonne estimation de la vitesse de l'anguille à tout instant.

L'oscillateur de Hopf proposé en tant que filtre nous permet donc de supprimer la composante sinusoïdale du signal pour ne garder que la valeur moyenne de celui-ci. Cependant, il est tout de même nécessaire d'avoir un ordre de grandeur de la dynamique d'évolution des signaux afin de choisir au mieux les paramètres de réglage de l'oscillateur.

5.4 Commande en boucle fermée

Ce paragraphe a pour but de présenter quelques lois de commande afin de réaliser des simulations de nages dans lesquelles l'anguille se dirige de manière autonome vers une position 3D désirée. Par autonome, on sous-entend que l'anguille est capable de stabiliser son angle de roulis et de contrôler son cap ainsi que son altitude pour pouvoir piloter la position de sa tête. Pour ce genre de robot, les différents travaux menés sur les stratégies de commande [McIsaac et Ostrowski, 2000, Morgansen *et al.*, 2001, Mason, 2003, Yu *et al.*, 2003, Aubin *et al.*, 2005, Lapierre et Jouvencel, 2005, Matsuno et Sato, 2005, Alamir *et al.*, 2007] ont montré que la compréhension du comportement dynamique est extrêmement important pour pouvoir obtenir des lois de commande efficaces. Pour l'ensemble des simulations en boucle fermée, on utilisera le générateur de mouvements à base de CPGs. Les ailerons présentés au chapitre 4, seront utilisés pour stabiliser le roulis.

5.4.1 Asservissement en cap

L'asservissement en cap de l'anguille consiste à écrire une loi de commande agissant uniquement sur la courbure plane de l'anguille via les termes A_L et A_R du générateur de mouvements à base de CPGs. Dans ce cas, on suppose que l'anguille évolue dans un plan (quelconque) et que, pour toutes les articulations rotules, les angles de roulis θ_k et de lacet ϕ_k sont nuls. Pour pouvoir asservir le cap de l'anguille vers un point cible situé dans l'espace, nous appliquons la stratégie proposée dans [Alamir *et al.*, 2007] et représentée sur la figure (5.17) :

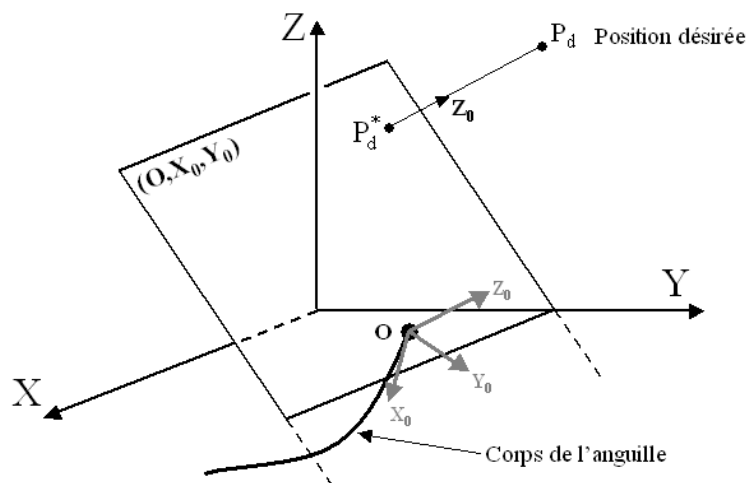


Figure 5.17: – Stratégie de contrôle pour le cap de l'anguille

On considère qu'à un instant t de la simulation l'anguille évolue dans un plan défini par le repère (O, X_0, Y_0) de la tête de l'anguille. Si le point cible P_d que l'on souhaite atteindre

n'est pas situé sur ce plan, on calcule la projection P_d^* de P_d sur le plan selon l'axe \mathbf{Z}_0 de la tête de l'anguille. Les auteurs proposent alors d'appliquer la méthode suivante : premièrement, on calcule les éléments qui suivent :

$$P_{sc} = -\overrightarrow{OP_d^*} \cdot \mathbf{x}_0 \quad ; \quad P_{ext} = -\pi_3 \left(\overrightarrow{OP_d^*} \wedge \mathbf{x}_0 \right) \quad (5.36)$$

où $\pi_3(\cdot)$ désigne la troisième composante du vecteur désigné.

Puis on détermine l'offset à appliquer sur les articulations par :

$$O_{ffs}(t) = \begin{cases} \beta_{\max} \cdot \overline{P}_{ext} & \text{si } \overline{P}_{sc} > 0 \\ \beta_{\max} & \text{sinon} \end{cases} \quad (5.37)$$

où \overline{P}_{ext} et \overline{P}_{sc} représentent les moyennes temporelle sur une période T de P_{ext} et P_{sc} .

Dans cette méthode, \overline{P}_{sc} permet de déterminer si le point est devant la tête (>0), ou en arrière de la tête (<0). L'élément P_{vc} permet de déterminer l'angle entre la direction courante de l'anguille et le vecteur $\overrightarrow{OP_d^*}$.

Pour les simulations qui vont suivre, nous adaptons la loi de commande de [Alamir *et al.*, 2007] dans le cadre bio-mimétique que nous nous sommes fixés dans ce chapitre. Plutôt que de calculer la moyenne temporelle des éléments P_{ext} et P_{sc} (équation 5.36), nous utilisons l'information de cap ψ_c de l'anguille obtenue à partir de l'oscillateur de Hopf proposé dans le paragraphe 5.3. La stratégie adoptée est la suivante :

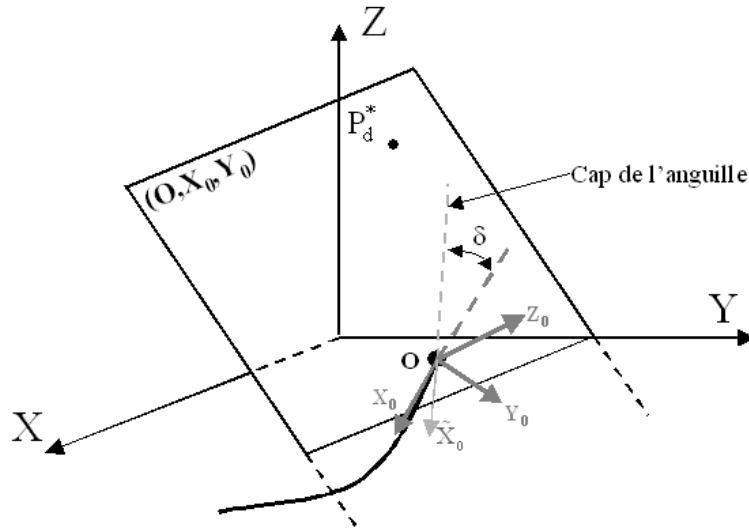


Figure 5.18: – Représentation du vecteur de cap $\tilde{\mathbf{x}}_0$ de l'anguille

Sur la figure (5.18), le vecteur $\tilde{\mathbf{x}}_0$ représente le vecteur de cap de l'anguille. Ce vecteur est obtenu par la rotation d'angle δ du vecteur \mathbf{x}_0 autour de l'axe \mathbf{z}_0 . On définit l'angle δ

comme la différence entre l'angle de cap ψ_c et l'angle de lacet de la tête ψ_0 .

La loi de commande est alors :

$$\delta = \psi_c - \psi_0 \quad (5.38)$$

$$\tilde{\mathbf{x}}_0 = \mathbf{rot}(\mathbf{z}_0, \delta) \cdot \mathbf{x}_0 \quad (5.39)$$

$$P_{sc} = -\overrightarrow{OP_d^*} \cdot \tilde{\mathbf{x}}_0 \quad (5.40)$$

$$P_{vc} = -\pi_3 \left(\frac{\overrightarrow{OP_d^*} \wedge \tilde{\mathbf{x}}_0}{\|\overrightarrow{OP_d^*}\|} \right) \quad (5.41)$$

Puis on calcule l'offset qui permet de faire tourner l'anguille :

$$O_{\text{ffs}}(t) = \begin{cases} \beta_{\text{max}} \cdot P_{vc} & \text{si } P_{sc} > 0 \\ \beta_{\text{max}} & \text{sinon} \end{cases} \quad (5.42)$$

où β_{max} représente l'angle maximum de courbure que l'on souhaite appliquer pour chaque liaison rotule.

Pour rappel, à partir de l'offset que l'on vient de calculer, on applique les relations suivantes afin d'agir sur les paramètres du générateur de mouvements à base de CPGs :

$$A_L = \frac{f(t) \cdot A_{LR}}{2} + \frac{O_{\text{ffs}}}{2} \quad (5.43)$$

$$A_R = \frac{f(t) \cdot A_{LR}}{2} - \frac{O_{\text{ffs}}}{2} \quad (5.44)$$

avec A_{LR} l'amplitude souhaitée de l'ondulation du corps.

Pour les simulations qui vont suivre, on considère que sur la longueur de l'anguille, on autorise une courbure maximum de 50° soit pour chaque articulation :

$$\beta_{\text{max}} = \frac{50^\circ}{N} \text{ avec } N=12, \text{ le nombre de liaisons rotules du robot.}$$

Pour tester le contrôleur de cap, nous réalisons une simulation dans laquelle l'anguille aura deux points successifs à rejoindre. Cette simulation est effectuée dans le plan $\mathbf{x}_w - \mathbf{y}_w$ du repère fixe et les deux points cibles sont définis de cette manière :

$$P_d = \begin{cases} (-10, -5, 0) & \text{pour } 0 \leq t < 85s \\ (-6, 0, 0) & \text{pour } t > 85s \end{cases}$$

Pour $t=0$, la configuration initiale de l'anguille est définie par :

- le repère de la tête Σ_0 est confondu avec le repère fixe Σ_w ;
- la vitesse initiale de l'anguille est nulle ;
- l'anguille est droite : $\forall k, \theta_k = \psi_k = \phi_k = 0$.

Les paramètres de nage pris pour cette simulation sont les mêmes que précédemment :

$$A_{LR}=0,15 \text{ rad}; \alpha=0,8 \text{ m}^{-1}; \Delta\phi = \frac{2\pi L_c}{\lambda}; L_c=0,14 \text{ m}; \lambda=1,5 \text{ m}; v=0.33 \text{ Hz}; t_f=4 \text{ s};$$

$$\phi_c = 0; \theta_c = 0$$

De plus, on impose que les ailerons restent horizontaux tout au long de la simulation.

La figure (5.19 -a) montre la position de la tête pendant la simulation, tandis que la figure (5.19 -b) représente la consigne de courbure O_{ffs} calculée par l'équation (5.42) :

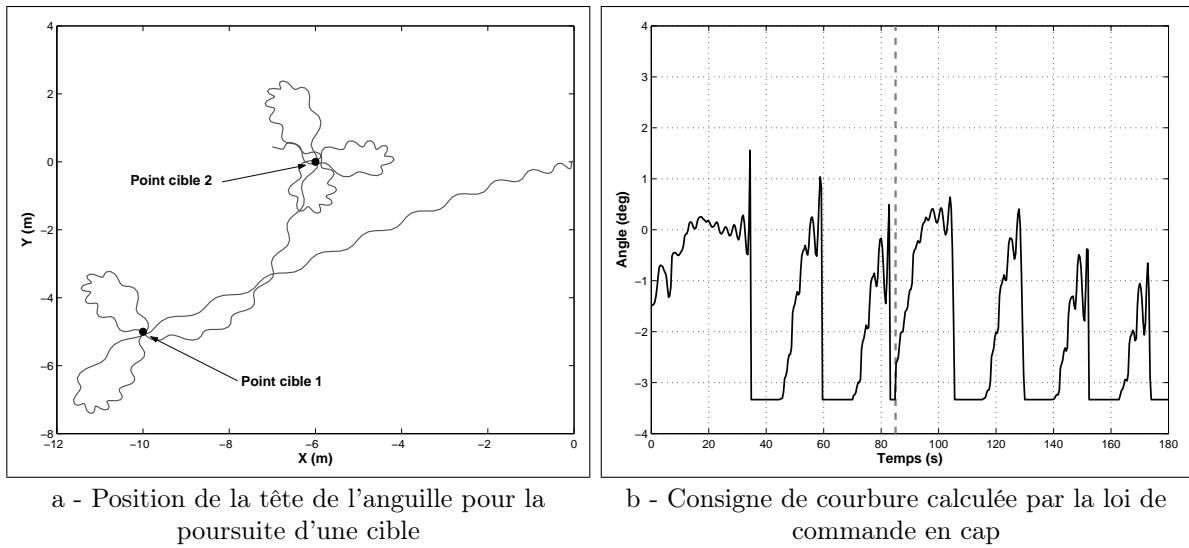


Figure 5.19: – Application du contrôleur de cap de l'anguille pour de la poursuite de cibles

L'analyse de la figure (5.19 -a) permet de voir que l'anguille essaye en permanence de rejoindre le point cible courant. Ainsi, on remarque qu'à partir de $t=85s$, l'anguille change de direction et se dirige vers le point cible N°2. On peut également observer qu'une fois le point cible dépassé, l'anguille se met à tourner toujours dans le même sens. Ce phénomène est directement lié à la loi de commande utilisée : en effet, dans l'équation (5.42), lorsque le produit scalaire entre $-\overrightarrow{OP_d^*}$ et $\tilde{\mathbf{x}}_0$ est négatif (c'est-à-dire lorsque l'anguille a dépassé le point cible), on applique un angle constant sur la courbure de l'anguille ce qui a pour conséquence de la faire tourner (ici vers la gauche).

On peut confirmer cette analyse en regardant la figure (5.19 -b). En effet, on aperçoit régulièrement sur cette courbe une saturation basse qui correspond au moment où le produit scalaire est négatif et indique que l'anguille a dépassé le point cible. Pour $t=85s$ (ligne verticale en pointillé) sur la figure (5.19 -b), on remarque que la durée de saturation est plus courte que dans le reste du graphique. Ceci est dû au fait qu'à ce moment, le point cible change, qu'il se trouve de côté par rapport à la tête de l'anguille, ce qui provoque un changement de signe immédiat dans le produit scalaire.

5.4.2 Asservissement en altitude

Pour réaliser l'asservissement en altitude de l'anguille, on a choisi dans un premier temps d'utiliser uniquement les deux ailerons et de fixer les consignes en tangage ϕ_k des articulations à zéro. Comme pour l'asservissement en cap, la stratégie de commande repose sur la figure (5.17). Dans le paragraphe précédent, on s'est attaché à ce que l'anguille se dirige vers le point P_d^* situé dans son plan de nage. Pour le contrôle en altitude, on fait varier le plan de nage $(\mathbf{O}, \mathbf{X}_0, \mathbf{Y}_0)$ afin que les points P_d et P_d^* deviennent confondus. Pour cela, nous proposons une loi de commande (différente de celle proposée dans [Alamir *et al.*, 2007]) :

1. On calcule la distance entre les points P_d et P_d^* en utilisant la relation :

$$P_{hsc} = \mathbf{P}_d^* \mathbf{P}_d \cdot \mathbf{z}_0 \quad (5.45)$$

2. Les consignes appliquées aux ailerons sont données par les lois de commande suivantes :
 - pour la commande en altitude :

$$q_a = -q_{a,\max} \tanh(k_3 \cdot P_{hsc}) \quad (5.46)$$

- pour le contrôle du roulis :

$$q_d = -q_{d,\max} \tanh(k_1 \phi - k_2 \omega_{x,0}) \quad (5.47)$$

où :

- $q_{a,\max}$ représente l'angle maximum appliqué aux ailerons pour la commande en altitude, et k_3 une constante de réglage.
- $q_{d,\max}$ représente l'angle maximum appliqué aux ailerons pour le contrôle du roulis, et (k_1, k_2) sont des constantes de réglage.

Pour rappel, les positions articulaires des ailerons sont définies par la somme de la consigne différentielle q_d permettant d'asservir l'angle de roulis et du couplage additionnel q_a pour régler l'altitude, soit :

$$q_{3n+1} = q_d - q_a \quad (5.48)$$

$$q_{3n+2} = q_d + q_a \quad (5.49)$$

(On rappelle que le calcul de q_d est présenté au chapitre 4).

Pour tester le commande en altitude, nous réalisons une nouvelle simulation dans laquelle l'anguille aura deux points successifs à rejoindre et définis par :

$$P_d = \begin{cases} (-10, -5, 5) & \text{pour } 0 \leq t < 100s \\ (-6, 0, 2) & \text{pour } t > 100s \end{cases}$$

Les paramètres de nage utilisés dans cette simulation sont les suivants :

$$A_{LR}=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; \Delta\phi = \frac{2\pi L_c}{\lambda} ; L_c=0,14 \text{ m} ; \lambda=1,5 \text{ m} ; v=0.33 \text{ Hz} ; t_f=4 \text{ s} ;$$

$$\phi_c = 0 ; \theta_c = 0, q_{a,\max}=15^\circ ; q_{d,\max}=30^\circ ; k_3=0,6$$

Les figures (5.20 -a) et (5.20 -b) représentent la position de la tête selon deux vues différentes :

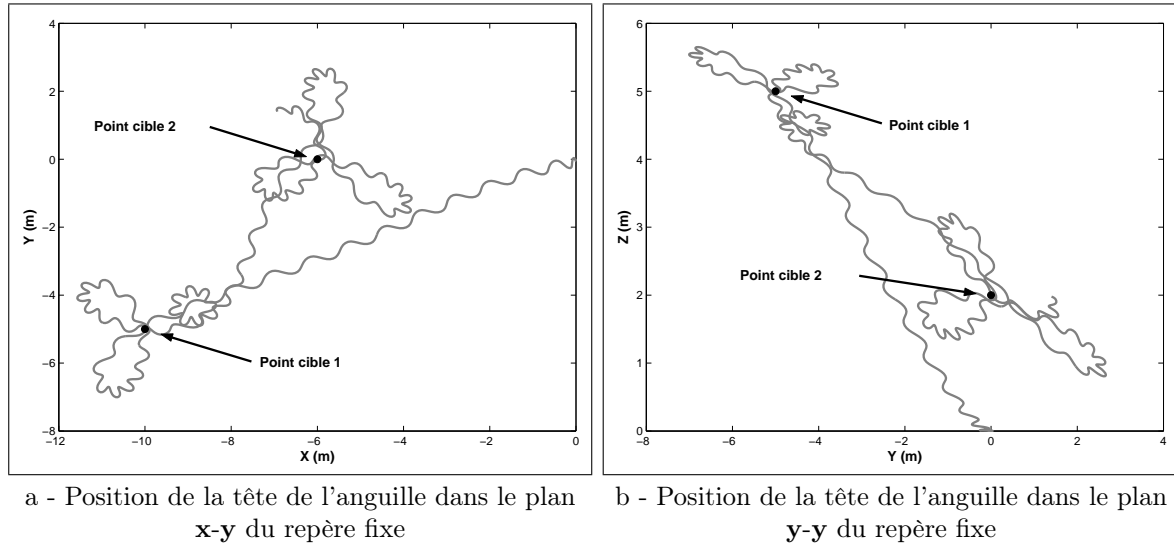


Figure 5.20: – Application des commandes de cap et d'altitude pour la poursuite de cibles 3D

Sur la figure (5.20 -a), on retrouve une courbe très similaire à la figure (5.19 -a), ce qui permet de valider le suivi de cap de l'anguille même lorsque l'anguille évolue dans un espace 3D. La figure (5.20 -b) permet d'analyser le comportement du réglage d'altitude. On remarque tout de suite qu'une fois arrivée au point cible, l'anguille tourne autour mais dans un plan qui n'est pas parallèle au plan (x - y) du repère fixe. Cela est dû au fonctionnement même de la commande en altitude : avec la méthode proposée l'anguille cherchera en permanence à avoir le point cible dans son plan de nage. Par conséquent, une fois que cela est fait, elle évoluera constamment dans ce plan. C'est donc pour cette raison que l'anguille tourne autour des points cibles, mais sur des plans d'évolution qui peuvent être différents.

La figure (5.21) présente l'angle de roulis de la tête de l'anguille au cours du temps. On remarque que le débattement est relativement faible $\pm 2^\circ$ et que le correcteur de roulis permet la diminution des oscillations jusqu'à ce que celles-ci deviennent nulles. Pour rappel, ce correcteur est très important, puisqu'il évite à l'anguille de tourner sur elle-même ce qui facilitera la commande pour un opérateur extérieur.

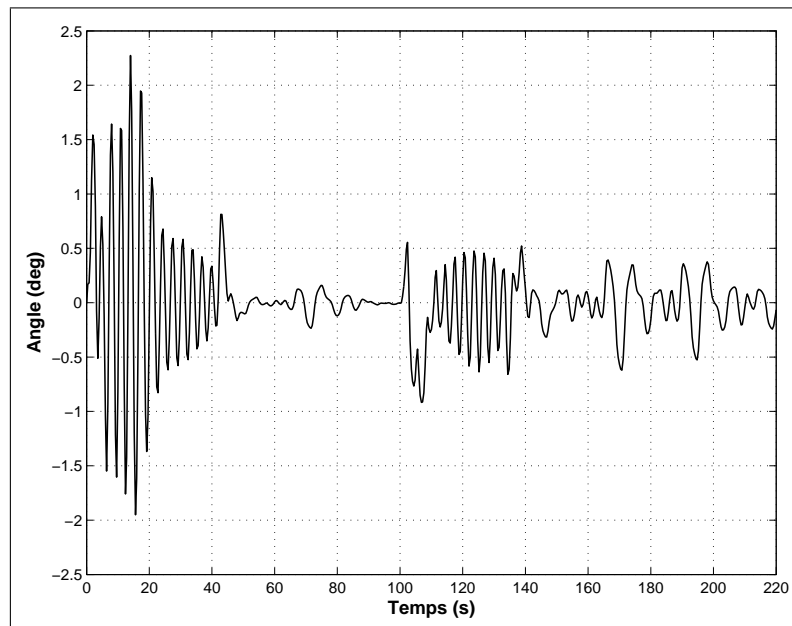


Figure 5.21: – Angle de roulis de la tête

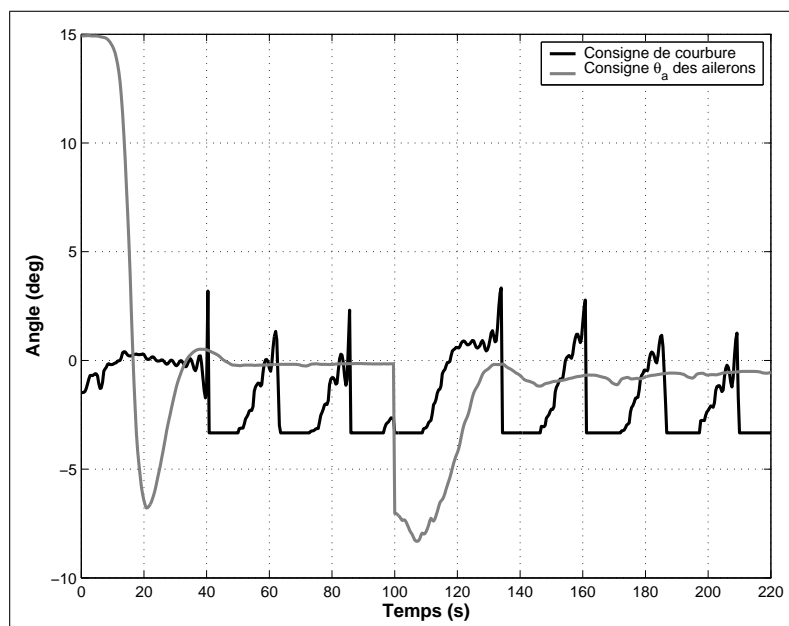


Figure 5.22: – Consignes calculées par les commandes de cap et d'altitude

La figure (5.22) permet d'observer les consignes calculées par la commande de cap et le contrôleur d'altitude. Pour la consigne de commande en cap, on retrouve une courbe similaire à celle calculée sur la figure (5.19-b). On y retrouve les saturations à $-\beta_{\max}$ qui indiquent que l'anguille est courbée au maximum afin d'effectuer un demi-tour pour revenir vers le point cible qu'elle vient de dépasser (cela se traduit par les boucles sur les trajectoires de la figure (5.20)). Pour la courbe de la consigne envoyée aux ailerons, on observe que le contrôleur essaye d'asservir les ailerons sur un angle de 0° : en d'autres termes, le contrôleur essaye de ramener le point cible désiré P_d dans le plan de nage de l'anguille. On peut également remarquer qu'il existe un écart statique entre l'angle de référence (0°) et l'angle auquel le correcteur arrive à asservir. Cela peut être partiellement réglé en augmentant le coefficient k_3 du régulateur (la commande sera plus énergique). De plus, il faudrait certainement ajouter un terme intégral dans la loi de commande afin d'être certain d'annuler cet écart statique.

A présent, nous réalisons la même simulation, mais en actionnant l'anguille différemment. Jusqu'ici, nous avons utilisé les ailerons pour la commande en altitude et le contrôle du roulis, et nous avons fixé à zéro les angles de tangage ϕ_k et de roulis θ_k . Cette fois, nous utilisons uniquement les ailerons pour le contrôle du roulis, et la commande d'altitude est appliquée sur les angles de tangage ϕ_k des liaisons rotules. Pour ce faire, nous utilisons une loi de commande similaire :

1. On calcule la distance entre les points P_d et P_d^* en utilisant la relation :

$$P_{hsc} = \mathbf{P}_d^* \mathbf{P}_d \cdot \mathbf{z}_0 \quad (5.50)$$

2. La consigne de tangage appliquée aux liaisons rotules :

$$\phi_c = \phi_{c,\max} \tanh(k_4 \cdot P_{hsc}) \quad (5.51)$$

3. Les consignes appliquées aux ailerons sont données par les lois de commande suivantes :

- la commande en altitude n'étant plus appliquée aux ailerons, on a alors :

$$q_a = 0 \quad (5.52)$$

- on garde cependant le contrôle du roulis soit :

$$q_d = -q_{d,\max} \tanh(k_1 \phi - k_2 \omega_{x,0}) \quad (5.53)$$

où $\phi_{c,\max}$ représente l'angle maximum appliqué pour l'angle de tangage des liaisons rotules et k_4 une constante de réglage.

Pour la simulation, on considère que l'anguille ne peut pas se courber au-delà de 40° en tangage sur la totalité de son corps. On définit alors $\phi_{c,\max}$ tel que :

$$\phi_{c,\max} = \frac{40^\circ}{N} \text{ avec } N=12, \text{ le nombre de liaisons rotules du robot.}$$

Les paramètres de nage utilisés dans cette simulation sont :

$$A_{LR}=0,2 \text{ rad} ; \alpha=0,8 \text{ m}^{-1} ; \Delta\phi = \frac{2\pi Lc}{\lambda} ; Lc=0,14 \text{ m} ; \lambda=1,5 \text{ m} ; v=0.33 \text{ Hz} ; t_f=4 \text{ s} ;$$

$$\theta_c = 0, q_{d,\max}=30^\circ ; k_4=0,6$$

Les figures (5.23 -a) et (5.23 -b) représentent la position de la tête selon deux vues différentes :

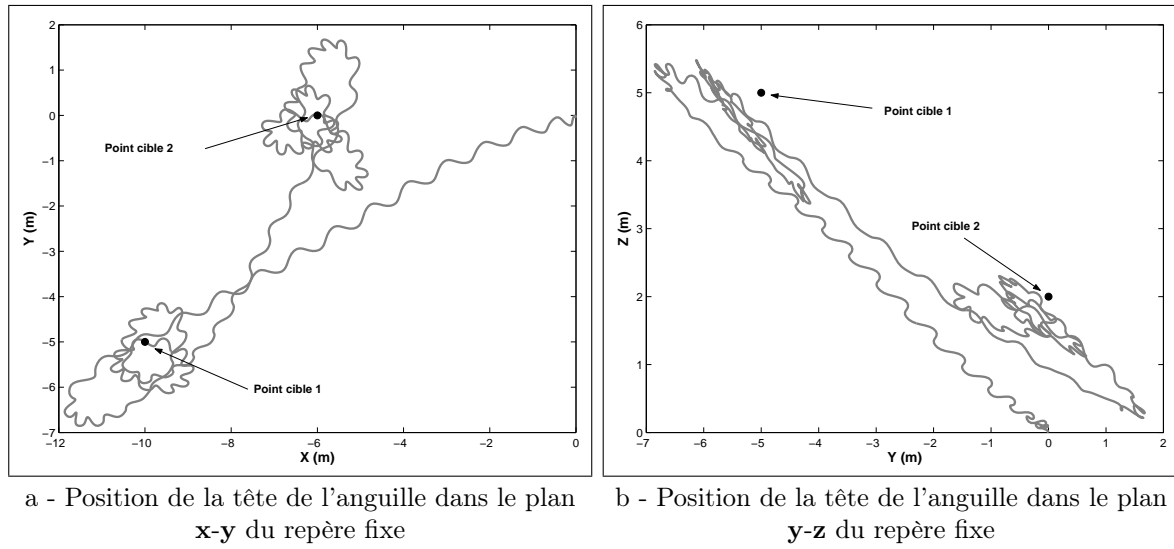


Figure 5.23: – Loi de commande pour la poursuite de cibles 3D en utilisant l'angle de tangage des liaisons rotules

Sur la figure (5.23 -a), on remarque que le contrôleur de cap fonctionne toujours et que l'anguille tourne bien autour du point cible. Cependant, sur la figure (5.23 -b), on remarque que le contrôleur d'altitude donne un résultat relativement mauvais. Même si l'anguille évolue bien dans un plan lorsqu'elle tourne autour du point cible, on peut observer qu'il y a un écart en altitude non négligeable entre ce plan et la cible. Ceci s'explique par le fait que le couplage entre le mouvement de nage et la courbure en tangage implique un mouvement de roulis trop important que le contrôleur n'arrive pas à stabiliser. On peut facilement s'en rendre compte en traçant l'angle de roulis de l'anguille au cours du temps (voir figure 5.24) :

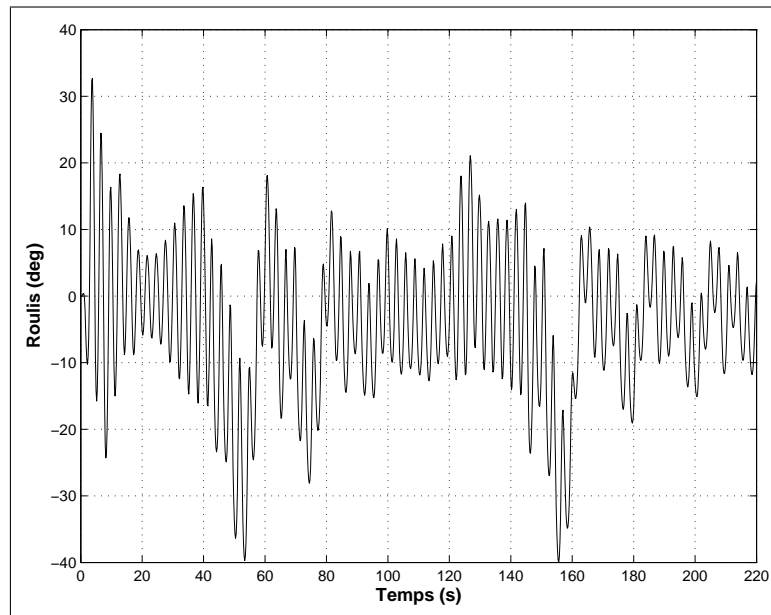


Figure 5.24: – Angle de roulis lorsqu'on utilise les angles de tangage des liaisons rotules

On remarque sur la figure (5.24), que l'angle de roulis est très important (jusqu'à 40°), comparé à la simulation dans laquelle on utilisait uniquement les ailerons. Ce large débattement empêche le correcteur d'altitude de fonctionner correctement, puisque la distance calculée entre les points P_d et P_d^* est constamment perturbée par le mouvement oscillant de la tête de l'anguille. Avec le correcteur utilisé pour les ailerons, on observe qu'il est tout juste possible de stabiliser l'angle à $\pm 10^\circ$, ce qui s'avère insuffisant pour cette simulation.

De ces deux simulations, on peut en conclure qu'il est préférable de faire du contrôle d'altitude en utilisant les ailerons plutôt que de courber le corps de l'anguille.

5.5 Commande avec perception de l'environnement

Dans les paragraphes précédents, nous avons étudié des lois de commande en boucle fermée (asservissement sur un objectif), sans se préoccuper des différents obstacles qui peuvent exister. Dans ce paragraphe, nous allons simplifier le problème en se limitant à la nage plane, et nous faisons évoluer l'anguille dans un chemin étroit (le but étant d'éviter les parois). Nous supposons que la tête de l'anguille dispose d'un organe de perception afin de détecter les obstacles.

5.5.1 Description du détecteur d'obstacles

Le détecteur d'obstacle, ou plutôt sonar, que nous proposons, est constitué de quatre capteurs de distance C_i ($i=1,\dots,4$) orientés selon plusieurs directions (voir figure 5.25) :

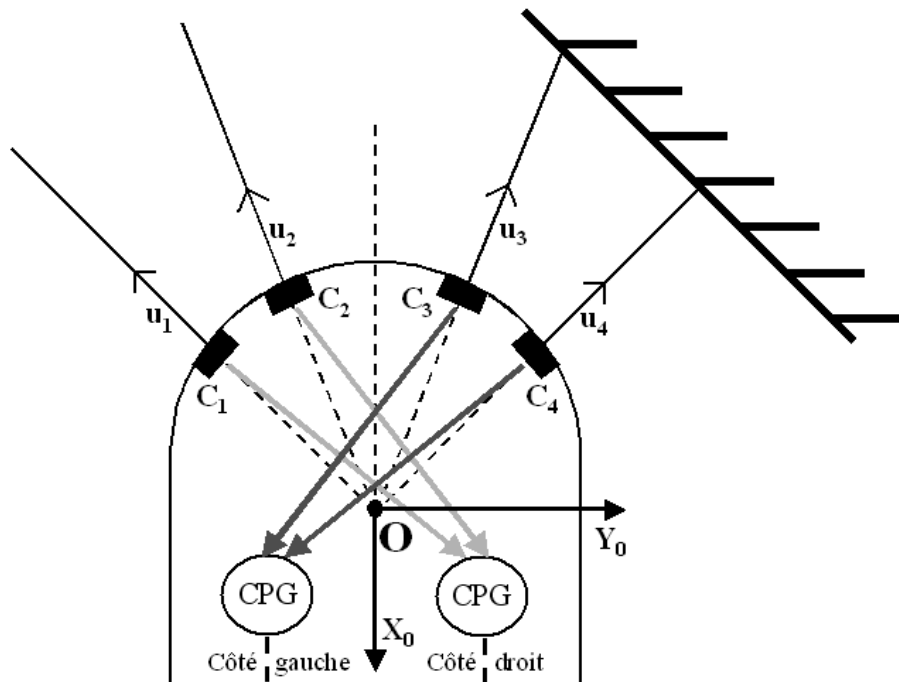


Figure 5.25: – Positionnement des capteurs par rapport à la tête de l'anguille

Les quatre capteurs sont orientés de $\pm\pi/4$ pour les capteurs (C_1, C_4) et $\pm\pi/8$ pour les capteurs (C_2, C_3) par rapport à l'axe \mathbf{x}_0 . De plus, les axes de détection de ces quatre capteurs se croisent au centre du repère Σ_0 . On supposera par la suite que ces capteurs ne sont pas capables de discerner un obstacle au-delà d'une certaine distance que l'on nommera d_{\max} .

Les capteurs sont ensuite regroupés par paires : les capteurs (C_1, C_2) pour détecter les obstacles à gauche de l'anguille et les capteurs (C_3, C_4) pour détecter les obstacles à droite de l'anguille.

On a vu au début de ce chapitre que le générateur de mouvements des articulations de l'anguille est composé de deux chaînes de CPGs : une chaîne pour les muscles du côté gauche et une chaîne pour le côté droit. Ainsi, on a pu voir que pour faire tourner l'anguille (vers la droite par exemple), il suffit d'augmenter les oscillations du côté gauche du corps et de diminuer les oscillations du côté droit. Pour retrouver ce mode de fonctionnement avec les capteurs, nous allons réaliser une connexion croisée entre les capteurs et les chaînes de CPGs : les capteurs du côté droit seront reliés aux CPGs de gauche et inversement. Ainsi, si les capteurs droits détectent un mur, alors ils devront diminuer l'amplitude des CPGs du côté gauche.

5.5.2 Loi de commande proposée

En considérant les capteurs liés à la tête de l'anguille, celle-ci ayant un mouvement d'oscillation permanent en rotation autour de l'axe \mathbf{z}_0 pendant la nage, les données de distance des obstacles risquent d'être peu exploitables. Pour faire en sorte que les axes de détection des capteurs restent relativement stables pendant la nage de l'anguille, nous allons utiliser l'oscillateur de Hopf proposé dans le paragraphe 5.3.

On considère à présent que les capteurs sont fixés sur un dispositif rotatif (le centre de rotation étant confondu avec le centre du repère Σ_0), dont l'axe médian est asservi sur l'axe du cap estimé de l'anguille. La procédure pour calculer les axes des capteurs est donc :

1. A partir de l'angle de lacet de la tête $\psi_0(t)$ de l'anguille, on estime le cap $\tilde{\psi}_0(t)$ en utilisant le filtre de Hopf ;
2. On définit alors les quatre axes des capteurs par :

$$\mathbf{u}_i = \mathbf{Rot}(\mathbf{z}_0, \tilde{\psi}_0) \cdot \mathbf{Rot}(\mathbf{z}_0, \alpha_i) \cdot \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad (5.54)$$

où la première rotation d'angle $\tilde{\psi}_0$ permet de déterminer l'axe médian des capteurs, et la seconde rotation d'angle α_i permet de définir l'axe du capteur C_i .

et α_i les coefficients définis par :

$$\alpha_1 = \frac{\pi}{4} ; \alpha_2 = \frac{\pi}{8} ; \alpha_3 = -\frac{\pi}{8} ; \alpha_4 = -\frac{\pi}{4}$$

A partir de l'axe \mathbf{u}_i de chaque détecteur C_i , on détermine la distance d_i du plus proche obstacle (si celui-ci existe et se trouve à une distance inférieure à d_{\max}).

La loi de commande que nous proposons est alors :

on calcule la moyenne des capteurs de gauche (C_1, C_2) et la moyenne des capteurs de droite (C_3, C_4), soit :

$$d_{\text{mg}} = \frac{1}{2}(d_1 + d_2) \quad d_{\text{md}} = \frac{1}{2}(d_3 + d_4) \quad (5.55)$$

On détermine enfin les consignes d'amplitude A_L et A_R que l'on doit appliquer aux chaînes de CPG de chaque côté de l'anguille par :

La figure (5.27) montre la position de la tête de l'anguille obtenu au cours de la simulation :

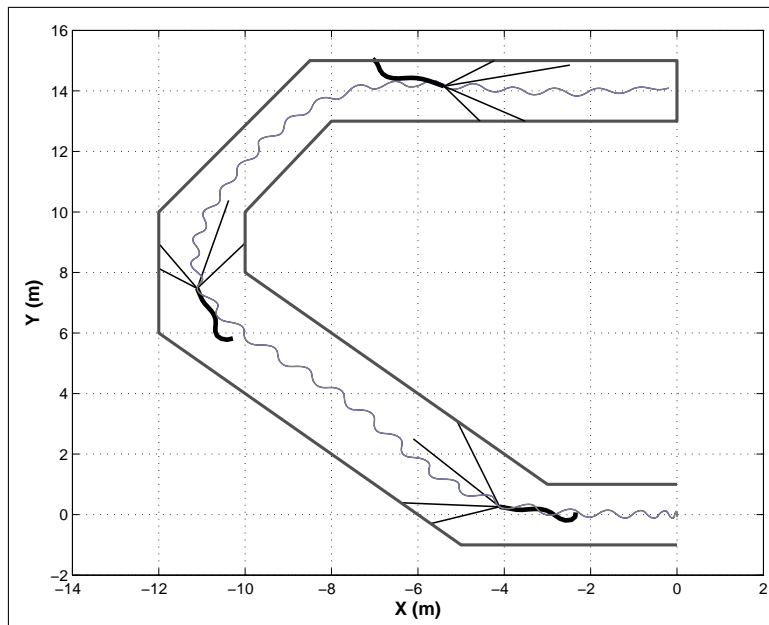


Figure 5.27: – Trajectoire de la tête et plusieurs configurations de l'anguille

On remarque que la tête de l'anguille ne sort pas de ce couloir et que, grâce aux capteurs de distance, la trajectoire se trouve en permanence (en dehors des transitoires) au centre des deux murs. Cependant, comme on ne pilote ici que la tête de l'anguille, nous n'avons aucune garantie que le reste du corps n'entre pas en collision avec l'une des parois (voir dernière configuration de l'anguille en haut de la figure).

Dans la loi de commande proposée, on calcule les amplitudes d'oscillations A_L et A_R . Comme ces deux valeurs ne sont pas très parlantes et qu'on préfère visualiser l'amplitude de l'onde du corps A_{LR} et la rotation du corps O_{ffs} , on applique les fonctions de transformation suivantes (qui sont l'inverse des équations 5.19 et 5.20) :

$$A_{LR} = A_L + A_R \quad (5.58)$$

$$O_{ffs} = A_L - A_R \quad (5.59)$$

Ces deux quantités calculées à partir des consignes A_L et A_R sont tracées sur la figure (5.28) :

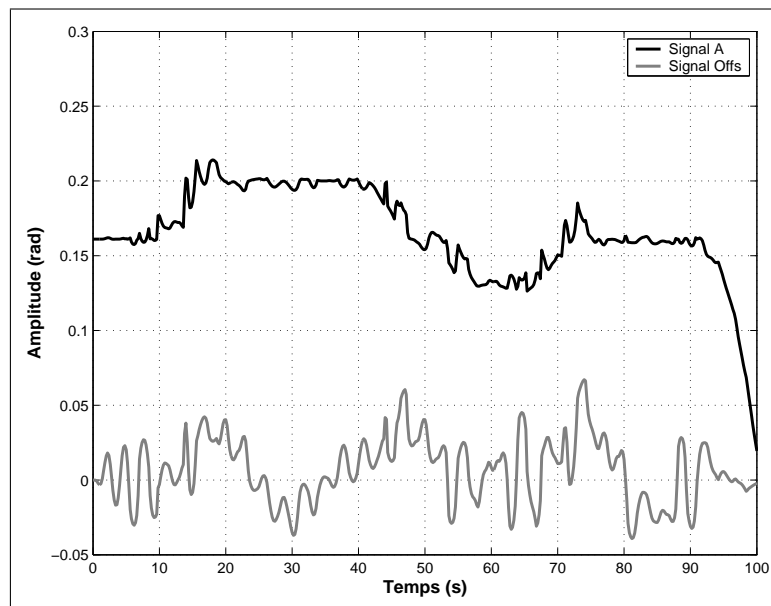


Figure 5.28: – Consignes calculées par la loi de commande

Sur la figure (5.28), on peut remarquer que la courbe de consigne d'amplitude A possède plusieurs paliers. En réalité, chacun de ces paliers correspond à un morceau du couloir. Ainsi, lorsque le couloir est large ($20 < t < 40s$), l'anguille augmente sa vitesse, et lorsque le couloir est étroit ($50 < t < 70s$) l'anguille la réduit. La courbe montrant l'offset ajouté à chaque articulation de lacet de l'anguille, montre que la loi de commande essaye en permanence d'asservir un cap parallèle à la direction des murs de chaque côté de l'anguille. Le caractère oscillant de cette courbe provient de l'oscillation en position de la tête. En effet, nous avons filtré la partie rotation de la tête (comme le font les yeux chez l'homme), mais il n'est pas possible d'annuler le mouvement transversal. Cependant, cette consigne étant envoyée aux CPGs qui possèdent un temps de convergence assez lent, on peut estimer que cette consigne sera naturellement filtrée.

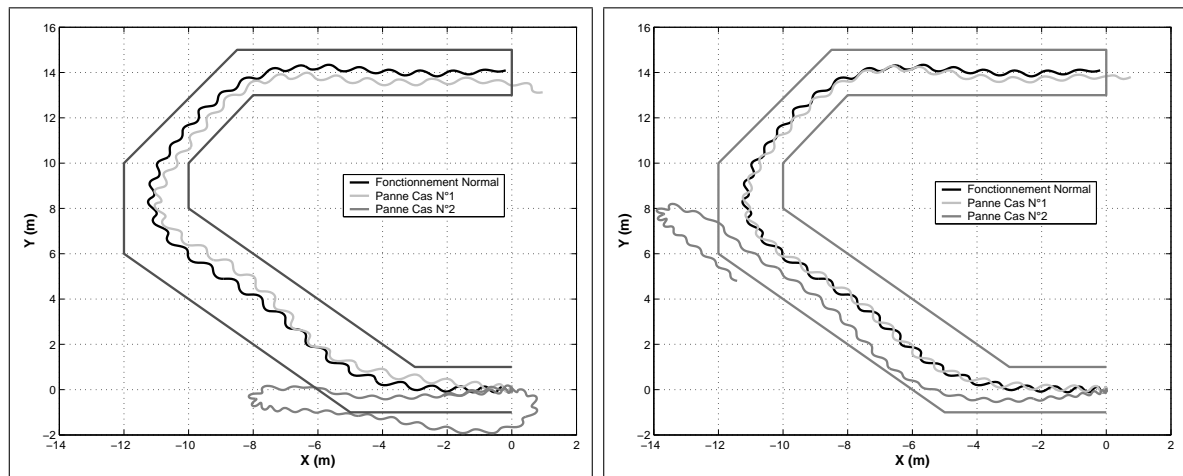
5.5.4 Simulations avec un ou plusieurs capteurs défectueux

A présent, on considère qu'un ou plusieurs détecteurs sont en panne, ce qui va permettre d'analyser le comportement de l'anguille dans cette situation. Pour cela, on va considérer deux cas de panne pouvant survenir :

- cas de panne n°1 : le capteur ne voit jamais les obstacles (la distance perçue est sa distance maximale) ;
- cas de panne n°2 : le capteur voit toujours un obstacle (la distance perçue est égale à zéro).

Pour voir l'influence de ces pannes sur les différents capteurs, on simule les pannes suivantes : sur la figure (5.29 -a), on simule les deux cas de panne affectant le capteur C_4

uniquement. Sur la figure (5.29-b) on simule également les deux cas de panne mais pour le capteur C_3 uniquement :



a - Panne sur un des capteurs externe

b - Panne sur un des capteurs interne

Figure 5.29: – Simulations de pannes sur les capteurs de l'anguille

Les résultats de la figure (5.29) montrent que le cas de panne n°2 est préjudiciable puisque l'anguille entre en contact avec le mur et sort du couloir. Par contre, le cas de panne n°1 ne pose pas de problème majeur, et on remarque que l'anguille nage plus près du mur du côté où le capteur est en panne. On peut également remarquer sur la figure (5.29-b) qu'une panne survenant sur un des capteurs au centre (C_2 ou C_3) ne modifie pas énormément la trajectoire de l'anguille : on peut donc en déduire que les capteurs externes sont prépondérants dans le fonctionnement de l'évitement d'obstacles.

Le cas de panne n°2 (qui nous donne de mauvais résultats) peut être assez facilement détecté par l'électronique du robot : on peut alors imaginer un réarrangement du comportement des capteurs en ne considérant plus celui qui est en panne.

On considère à présent qu'on utilise uniquement deux capteurs sur les quatre (ou que deux d'entre eux sont en panne). On effectue alors deux simulations en ne considérant que les capteurs (C_2, C_3) pour la première et les capteurs (C_3, C_4) pour la seconde. La figure (5.30) représente alors le résultat de ces deux simulations :

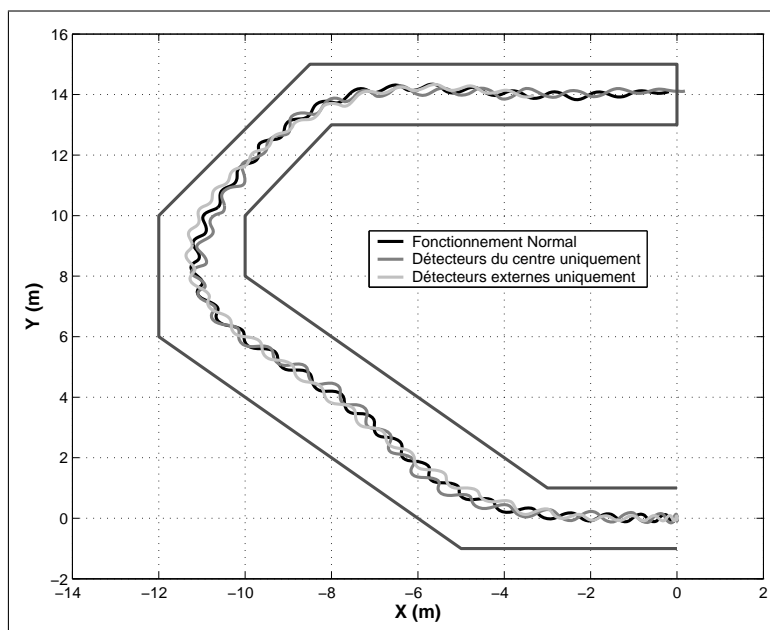


Figure 5.30: – Simulations avec seulement deux capteurs

Les deux simulations montrent que l'utilisation de deux capteurs symétriques suffit pour obtenir une trajectoire semblable à celle obtenue par la simulation avec les quatre capteurs. Seulement, pour le même temps de simulation, on remarque sur la figure (5.30) que l'anguille va plus ou moins loin. Ce phénomène est lié à la position de la paire de capteurs utilisée : pour les deux capteurs externes, les murs semblent "plus près" que pour les deux capteurs centraux (du fait de l'angle d'ouverture). Or, comme la distance globale perçue par les capteurs influe sur l'amplitude des ondulations, plus les murs seront près, plus l'anguille nagera lentement. C'est pourquoi avec les capteurs externes, l'anguille va moins loin qu'avec les capteurs centraux.

5.6 Conclusion

Dans ce chapitre, nous avons utilisé le modèle dynamique du prototype du robot anguille étudié dans le chapitre 4 et avons utilisé un générateur de mouvements basé sur une approche plus bio-mimétique. Nous avons vu que ce générateur à base de CPGs (qui sont équivalents à des réseaux de neurones) permet d'une part de reproduire la loi analytique utilisée dans les précédents chapitres, et d'autre part de s'adapter de manière autonome aux différents changements de consignes des paramètres de la loi d'ondulation (calculées par les lois de commande). Nous avons également proposé un filtre basé sur l'apprentissage des caractéristiques du signal d'entrée afin d'en déduire sa valeur moyenne. Enfin, nous avons simulé des lois de commande simples qui permettent de réaliser des commandes en boucle fermée de l'anguille. Ainsi, sur la base des travaux de [Alamir *et al.*, 2007] nous

avons simulé une première loi de commande dont le but est de permettre à l'anguille de rejoindre un point situé dans l'espace. Dans un second temps, nous avons proposé d'utiliser des capteurs de distance afin que l'anguille puisse se diriger dans un espace restreint.

Les résultats en boucle fermée que nous avons présentés utilisent des lois de commande simples. Il serait donc intéressant d'étudier d'autres lois de commande permettant d'améliorer la nage de l'anguille (loi de commande en vitesse par exemple). De même, la solution d'évitement d'obstacle est adaptée à un milieu dont les parois sont continues. Dans le cas de discontinuités avec des obstacles ponctuels, la loi de commande ne fonctionnerait pas convenablement. De plus, nous avons piloté la tête de l'anguille sans se soucier si le reste du corps pouvait entrer en contact avec les obstacles. Pour tenter de résoudre ce problème, une solution pourrait être d'utiliser les travaux de [Sfakiotakis *et al.*, 2006] dans lesquels les auteurs proposent d'ajouter des capteurs de distance non seulement sur la tête, mais aussi sur chaque vertèbre de l'anguille. Ainsi, l'anguille serait capable d'adapter chacune de ses vertèbres afin d'éviter un obstacle.

Dans la suite du projet Robot Anguille, dans le cadre du projet RAAMO de l'ANR, nous n'utiliserons pas de détecteurs de distance comme dans ce chapitre. Le détecteur qui sera développé sera basé sur le principe de l'électrolocation [Rasnow, 1996, Smith, 1999, Solberg *et al.*, 2007], c'est-à-dire la localisation d'obstacles par l'émission et la réception d'un champ électrique. Le principe de fonctionnement repose premièrement sur l'émission du champ électrique à partir d'organes dipolaires situés dans la tête et la queue de l'anguille. Ensuite, du fait de la différence de conductivité entre la peau de l'anguille et l'eau, les lignes de champ électrique sont attirées vers le corps (voir figure 5.31) :

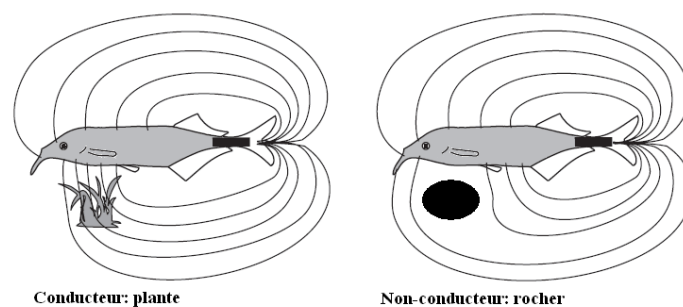


Figure 5.31: – Principe de l'électrolocation pour le poisson éléphant

A partir de l'image du champ électrique perçue par la peau du poisson et d'une image de référence, celui-ci est capable de déterminer la localisation et le type des obstacles. Dans le cas du prototype, cette méthode sera appliquée en équipant chaque vertèbre de détecteurs de champ électrique.

Conclusions et perspectives

Le travail présenté dans ce mémoire est consacré à la modélisation dynamique et la commande d'un robot anguille. Pour la partie modélisation dynamique, nous avons traité le cas d'un robot à architecture sérielle et le cas d'un robot hybride, c'est-à-dire constitué d'un ensemble de robots parallèles connectés en série. Pour ces deux robots, et suite aux travaux de [Ibrahim, 2006], nous avons proposé des algorithmes simples à écrire et à calculer que ce soit pour le modèle dynamique inverse ou le modèle dynamique direct. Pour cela, nous avons étendu les algorithmes de Newton-Euler au cas des robots où la base n'est pas fixe, en intégrant dans les calculs une étape permettant de déterminer les accélérations de la base (la tête de l'anguille dans le cas de notre robot).

L'étude du robot à structure sérielle, reprenant l'agencement des vertèbres de la colonne vertébrale de l'anguille, nous a permis de tester les nouveaux algorithmes de Newton-Euler sur un système relativement simple et d'obtenir des résultats préliminaires. Ainsi, en relation avec les choix technologiques pressentis pour le prototype réel, ce simulateur a permis d'une part, d'aider au dimensionnement des moteurs, et d'autre part de simuler les performances qu'aurait le robot avec les choix définitifs. La modélisation du prototype hybride, qui représente la deuxième étape de ce mémoire, est basée sur les mêmes principes que le robot série mais présente une philosophie différente dans sa structure. Ce robot étant constitué de plusieurs modules parallèles, le calcul dynamique de l'anguille passe par la modélisation dynamique de ceux-ci avant de passer au robot en entier. Ces calculs pouvant être coûteux lorsque le robot est constitué d'une multitude de modules, nous avons montré que dans le cas de notre robot, le modèle dynamique des modules pouvait être simplifié. En effet, nous avons remarqué que dans le cas où les paramètres inertiels des jambes sont négligeables devant ceux de la base et de la plate-forme, alors les modèles dynamiques s'y rapportant pouvaient être à leur tour négligés.

Outre la modélisation de la dynamique du robot, nous avons étudié un modèle de contact entre le fluide et la structure de l'anguille. Ce modèle basé sur les équations de Morison [Morison *et al.*, 1950] est un modèle très simple mais peu réaliste dans certains cas, surtout lors de virages dans la nage de l'anguille. En effet, le modèle suppose que la "peau" de l'anguille est complètement rigide entre deux articulations rotules, ce qui ne

sera pas le cas pour le prototype final. Malgré tout, ce modèle a permis de valider nos algorithmes, et de créer une première version de simulateur du prototype utilisable pour tester différentes lois de commande. Dans cette optique, nous avons présenté un travail préliminaire (qu'il faudrait pousser avec un modèle fluide plus performant) concernant l'influence des paramètres de la loi d'ondulation sur les performances en terme de vitesse d'avance et de couples demandés. Ainsi, nous avons pu voir que le problème est relativement complexe et qu'une méthode d'optimisation serait intéressante pour garantir une efficacité permanente de la nage. Pour finir, ce modèle fluide nous a permis de montrer que la présence d'ailerons au niveau de la tête de l'anguille est indispensable dès lors que l'on souhaite effectuer des nages en trois dimensions.

Pour pouvoir commander l'anguille et se placer dans un cadre bio-mimétique, nous avons utilisé comme générateur de mouvements un système composé d'un ensemble de CPGs modélisant le comportement des réseaux de neurones des poissons à nage anguilliforme. Pour le cas de l'anguille, nous avons pu voir que l'intérêt de ces générateurs de mouvements est leur capacité d'auto-adaptation aux différents signaux d'entrée. Sur cette base, nous avons présenté deux méthodes de commande dans le but de faire du contrôle en boucle fermée : une première méthode de commande qui permet de se diriger vers un point cible dans l'espace [Alamir *et al.*, 2007] et une deuxième méthode qui permet à l'anguille, par l'intermédiaire de capteurs de distance, de se diriger dans un couloir restreint. La première méthode proposée nous a permis de valider la possibilité de faire de la nage 3D et le fait qu'on puisse agir correctement sur les articulations pour atteindre un objectif. Cela valide également la possibilité de faire de la télé-opération, sachant que l'on confère à l'anguille une autonomie complète dans la génération de ses mouvements internes. La seconde méthode que nous avons proposée se base sur une philosophie différente : l'anguille ne cherche plus à se diriger vers une cible, mais essaye d'éviter les obstacles environnants. Pour cela, nous nous sommes basés sur la méthode de répulsion, méthode où les détecteurs agissent sur la courbure de l'anguille afin qu'elle s'éloigne des différents obstacles perçus (avec plus ou moins de force selon la distance).

La partie expérimentale du projet est toujours en cours, la partie mécanique de la structure des vertèbres étant terminée. Les boucles d'asservissement permettant de piloter les moteurs de l'anguille ont été testées et validées (voir annexe A), ainsi que les modèles géométriques inverses. N'ayant pas toutes les cartes électroniques, et la tête de l'anguille n'étant pas réalisée, nous n'avons pas encore testé l'ensemble complet. De même, le système de la peau n'étant pas tout à fait finalisé, nous n'avons pas caractérisé le comportement mécanique de celle-ci afin de l'intégrer dans le calcul du modèle dynamique.

Les perspectives liées à ce travail sont les suivantes :

- L'intégration d'un modèle fluide plus performant basé sur les travaux de [Porez, 2007]. Pour cela, nous allons devoir remplacer notre modèle de corps rigide entre deux liaisons rotules par une géométrie virtuelle : cette transformation permet de passer d'une anguille dont la géométrie est un ensemble de segments, à une géométrie où l'anguille est continue (et modélisant la fibre neutre que prendra la peau une fois sur le prototype). Seront ensuite appliquées les formules de calcul du modèle fluide développées dans cette thèse.
- Intégrer dans le simulateur un modèle de comportement de la peau afin de simuler l'énergie nécessaire pour la déformer.
- Transférer l'algorithme du modèle dynamique du prototype au laboratoire Gipsa-Lab de Grenoble afin qu'ils puissent tester leurs algorithmes de commande sur le simulateur, et ensuite les implanter directement dans le robot.
- Etendre la détection d'obstacles présentée dans le chapitre 5 à tout le corps de l'anguille (et pas seulement la tête) afin d'améliorer le comportement. On pourra également essayer de généraliser cette méthode pour la nage 3D, et ainsi l'intégrer dans les algorithmes de commande.
- Enfin, appliquer l'ensemble de ces méthodes au prototype et mettre en place la téléopération.

Bibliographie

- [Alamir *et al.*, 2007] M. Alamir, G. Hafidi, N. Marchand, M. ElRafei, M. Porez, et F. Boyer , « Feedback design for 3d movement of an eel like robot », In *IEEE Conference on Robotics and Automation, Rome, Italie*, pp. 256–261, 2007.
- [Aubin *et al.*, 2005] R. Aubin, P. Blazevic, et B. Clément , « Simulation of an novel snake-like robot », In *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR*, pp. 875–882, 2005.
- [Ayers *et al.*, 2000] J. Ayers, C. Wilbur, et C. olcott , « Lamprey robots », In *International Symposium on Aqua Biomechanisms*, 2000.
- [Barrett, 9996] D.S. Barrett , *Propulsive efficiency of a flexible hull underwater vehicle* , Thèse de Doctorat, Massachussetts Institute of Technology, 19996.
- [Boyer *et al.*, 2006a] F. Boyer, M. Alamir, D. Chablat, W. Khalil, A. Leroyer, et P. Lemoine , « Robot anguille sous-marin en 3d », In *Techniques de l'Ingénieur*. 2006.
- [Boyer *et al.*, 2006b] F. Boyer, M. Porez, et W. Khalil , « Macro-continuous torque algorithm for a three-dimensional eel-like robot », *IEEE Robotics transaction*, vol. 22, no. 4, pp. 763–775, 2006.
- [Chablat et Wenger, 2005] D. Chablat et P. Wenger , « Design of a spherical wrist with parallel architecture : Application to vertebrae of an eel robot », In *IEEE Conf. On Robotics and Automation, Barcelona, Spain*, pp. 3347–3352, April 2005.
- [Conradt et Varshavskaya, 2003] J. Conradt et P. Varshavskaya , « Distributed central pattern generator control for a serpentine robot », *Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003*, 2003.
- [Crespi et Ijspeert, 2006] A. Crespi et A.J. Ijspeert , « AmphiBot II : An amphibious snake robot that crawls and swims using a central pattern generator », In *Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, pp. 19–27, 2006.
- [Ekeberg, 1993] Ö. Ekeberg , « A combined neuronal and mechanical model of fish swimming », *Biological cybernetics*, vol. 69, pp. 363–374, 1993.
- [Featherstone, 1983] R. Featherstone , « The calculation of robot dynamics using articulated-body inertias », *Int. Journ. Of Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.

- [Fukunaga *et al.*, 2004] S. Fukunaga, Y. Nakamura, K. Aso, et S. Ishii , « Reinforcement learning for a snake-like robot controlled by a central pattern generator », In *IEEE Conference on Robotics, Automation and mehantronics*, volume 2, pp. 909–914, 2004.
- [Gallot *et al.*, 2007] G. Gallot, O. Ibrahim, et W. Khalil , « Dynamic modeling and simulation of a 3-d hybrid structure eel-like robot », In *IEEE International Conference on Robotics and Automation ICRA, Rome*, pp. 1486–1491, Avril 2007.
- [Galls, 2001] S.F. Galls , *Development of a computational model for an underwater autonomous vehicle* , Thèse de Doctorat, A&M University, Texas, 2001.
- [Gillis, 1998] G.B. Gillis , « Environmental effects on undulatory locomotion in the american eel *anguilla rostrata* : kinematics in water and on land », *J. of Experimental Biology*, vol. 201, pp. 949–961, 1998.
- [Gray, 1935] J. Gray , « The propulsive powers of the dolphin », *Journal of Experimental Biology*, pp. 192–199, Août 1935.
- [Hill, 1998] S.J. Hill , *Large amplitude fish swimming* , Thèse de Doctorat, University of Leeds, 1998.
- [Hu *et al.*, 2006] H. Hu, J. Liu, I. Dukes, et G. Francis , « Design of 3d swim patterns for autonomous robotic fish », In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, pp. 2406–2411, 2006.
- [Ibrahim, 2006] Ouarda Ibrahim , *Contribution à la modélisation dynamique des robots parallèles et des robots hybrides* , Thèse de Doctorat, Ecole Centrale / Université de Nantes, 2006.
- [Ijspeert et Crespi, 2007] A.J. Ijspeert et A. Crespi , « Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model », In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*, pp. 262–268, 2007.
- [Ijspeert *et al.*, 2007] A. Ijspeert, A. Crespi, D. Ryczko, et J.-M. Cabelguen , « From swimming to walking with a salamander robot driven by a spinal cord model », *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.
- [Inoue *et al.*, 2004] K. Inoue, S. Ma, et C. Jin , « Neural oscillator network-based controller for meandering locomotion of snake-like robots », In *IEEE International Conference on Robotics and Automation, New Orleans, LA*, April 2004.
- [Katz et Plotkin, 2002] J. Katz et A. Plotkin , *Low Speed Aerodynamics* Cambridge Univ. Press, second. ed., 2002.
- [Khalil et Creusot, 1997] W. Khalil et D. Creusot , « Symoro+ : a system for the symbolic modelling of robots », *Robotica*, vol. 15, pp. 153–161, 1997.
- [Khalil et Dombre, 2002] W. Khalil et E. Dombre , *Modeling identification and control of robots* Penton-Sciences, London, 2002.

- [Khalil *et al.*, 2007] W. Khalil, G. Gallot, et F. Boyer , « Dynamic modeling and simulation of a 3-d serial eel like robot », *IEEE Transaction on Systems, Man and Cybernetics Part C*, vol. 37, pp. 1259–1268, Novembre 2007.
- [Khalil *et al.*, 2005] W. Khalil, G. Gallot, O. Ibrahim, et F. Boyer , « Dynamic modeling of a 3-d serial eel-like robot », In *IEEE, International Conference on Robotics and Automation, Barcelone*, pp. 1282–1287, Avril 2005.
- [Khalil et Ibrahim, 2007] W. Khalil et O. Ibrahim , « General solution for the dynamic modeling of parallel robots », *Journal of Intelligent and robotic systems*, vol. 49, pp. 19–37, 2007.
- [Khalil et Kleinfinger, 1986] W. Khalil et J.-F. Kleinfinger , « A new geometric notation for open and closed-loop robots », In *Proc. IEEE Conf. on Robotics and Automation, San Francisco*, pp. 1174–1180, April 1986.
- [Khalil et Kleinfinger, 1987] W. Khalil et J.-F. Kleinfinger , « Minimum operations and minimum parameters of the dynamic model of tree structure robots », *IEEE J. of Robotics and Automation*, vol. RA-3(6), pp. 517–526, December 1987.
- [Khosla, 1986] P.K. Khosla , *Real-time control and identification of direct drive manipulators* , Thèse de Doctorat, Carnegie Mellon University, Pittsburgh, USA, 1986.
- [Kim et Youm, 2004] E.J. Kim et Y. Youm , « Design and dynamic analysis of fish robot : Potuna », In *IEEE Conf. On Robotics and Automation*, pp. 4887–4892, New Orleans, LA, April 2004.
- [Kleinfinger et Khalil, 1986] J.-F. Kleinfinger et W. Khalil , « Dynamic modelling of closed-chain robots », In *16th Int Symp on Industrial Robots. Bruxelles*, pp. 401–412, 1986.
- [Lachat *et al.*, 2006] D. Lachat, A. Crespi, et A.J. Ijspeert , « Boxybot : a swimming and crawling fish robot controlled by a central pattern generator », In *Proceedings of The first IEEE / RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2006)*, 2006.
- [Lapierre et Jouvencel, 2005] L. Lapierre et B. Jouvencel , « Path following control for an eel-like robot », In *Oceans 2005 - Europe*, volume 1, 2005.
- [Leroyer et Visonneau, 2005] A. Leroyer et M. Visonneau , « Numerical methods for ranse simulations of a self-propelled fish-like body », In SPE C. ISS, editor, *Journal of Fluids and Structures*, volume 20, Octobre 2005.
- [Lewis, 1989] E.V. Lewis , *Principles of naval architecture volume III motion in waves and controllability* The Society of Naval Architects and Marine Engineers, 1989.
- [Lighthill, 1971] J. Lighthill , « Large-amplitude elongated-body theory of fish locomotion », *Proc. R. Soc. Lond, Ser.*, vol. B179, pp. 125–138, 1971.

- [Liu et Hu, 2004] J. Liu et H. Hu , « A 3d simulator for autonomous robotic fish », *International Journal of Automation and Computing*, vol. 1, pp. 42–50, 2004.
- [Luh *et al.*, 1980] J.Y.S. Luh, M.W. Walker, et Paul R.C.P. , « On-line computational scheme for mechanical manipulator », *Trans. of ASME, J. of Dynamic Systems, Measurement, and control*, vol. 102(2), pp. 69–76, 1980.
- [Mason, 2003] R. Mason , *Fluid Locomotion and Trajectory Planning for Shape-Changing Robots* , Thèse de Doctorat, California Institute of Technology, Pasadena, California, 2003.
- [Mason et Burdick, 1999] R. Mason et J. Burdick , « Propulsion and control of deformable bodies in an ideal fluid », In *IEEE International Conference on Robotics & Automation, Detroit, Michigan*, pp. 773–780, May 1999.
- [Mason et Burdick, 2000] R. Mason et J.W. Burdick , « Experiments in caranguiform robotic fish locomotion », In *IEEE Int. Conf. Robotics and Automation*, pp. 428–435, 2000.
- [Matsuno et Sato, 2005] F. Matsuno et H. Sato , « Trajectory tracking control of snake robots based on dynamic model », In *IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pp. 3040–3045, Avril 2005.
- [McIsaac et Ostrowski, 1999] K.A. McIsaac et J.P. Ostrowski , « A geometric approach to anguilliform locomotion modelling of an underwater eel robot », In *IEEE Int. Conf. robotics and Automation*, pp. 2843–2848, 1999.
- [McIsaac et Ostrowski, 2000] K.A. McIsaac et J.P. Ostrowski , « Motion planning for dynamic eel-like robot », In *IEEE International Conference on Robotics and Automation, San Francisco*, pp. 1695–1700, April 2000.
- [Mihalachi, 2000] D. Mihalachi , *Système de commande temps réel distribué pour un mini robot autonome de type serpent* , Thèse de Doctorat, Université de Metz, 2000.
- [Müller *et al.*, 1997] U. K Müller, B.-L. E. van den Heuvel, E. J. Stamhuis, et J. J. Videler , « Fish foot prints : morphology and energetics of the wake behind a continuously swimming mullet (*chelon labrosus risso*) », *The Journal of Experimental Biology*, vol. 200, pp. 2893–2906, 1997.
- [Morgansen *et al.*, 2001] K.A. Morgansen, V. Duindam, R.J. Mason, J.W. Burdick, et Murray R.M. , « Nonlinear control methods for planar carangiform robot fish locomotion », In *IEEE International Conference on Robotics and Automation, Seoul, Korea*, pp. 427–434, May 2001.
- [Morison *et al.*, 1950] J.R. Morison, M.P. O'Brien, J.W. Johnson, et S.A. Shaaf , « The force exerted by surface waves on piles », *Transaction of the AIME*, vol. 189, pp. 149–154, 1950.

- [Nakada *et al.*, 2004] K. Nakada, T. Asai, et Y. Amemiya , « Design of an artificial central pattern generator with feedback controller », *Intelligent Automation and Soft Computing*, vol. 10, no. 2, pp. 185–192, 2004.
- [Nakamura et Ghodoussi, 1988] Y. Nakamura et M. Ghodoussi , « A computational scheme of closed link robot dynamics derived by d’alembert principle », In *Int. Conf. on robotics and automation, IEEE*, pp. 1354–1360, Philadelphia, 1988.
- [Or *et al.*, 2002] H. Or, J. Hallam, D. Willshaw, et A.J. Ijspeert , « Evolution of efficient swimming controllers for a simulated lamprey », In *From Animals to Animats, Proceedings of the 7th International Conference on the Simulation of Adaptive Behavior (SAB2002)*, 2002.
- [Ostrowski et Burdick, 1998] J.P Ostrowski et J. W. Burdick , « The geometric mechanics of undulatory robotics locomotion », *The international journal of robotics research*, vol. 17, no. 7, pp. 683–701, 1998.
- [Porez, 2007] M. Porez , *Modèle dynamique analytique de la nage tridimensionnelle anguilliforme pour la robotique* , Thèse de Doctorat, Ecole des Mines de Nantes / Université de Nantes, Septembre 2007.
- [Rasnow, 1996] B. Rasnow , « The effects of simple objects on the electric field of *Apteronotus* », *J. Comp. Physiol. A*, vol. 178(3), pp. 397–411, 1996.
- [Righetti *et al.*, 2005] L. Righetti, J. Buchli, et A.J. Ijspeert , « From dynamic hebbian learning for oscillators to adaptative central pattern generators », In *Proceeding of 3rd International Symposium on Adaptative Motion in Animals and machines - ANAM*, 2005.
- [Sfakiotakis *et al.*, 1999] M. Sfakiotakis, D.M. Lane, et J.B.C. Davies , « Review of fish swimming modes for aquatic locomotion », *IEEE Journal of oceanic engineering*, vol. 24, no. 2, pp. 237–252, April 1999.
- [Sfakiotakis *et al.*, 2007] M. Sfakiotakis, D.P. Tsakiris, et K. Karakasiliotis , « Polychaete-like pedundulatory robotic locomotion », In *IEEE international Conference on Robotics and Automation, Roma, Italy*, pp. 269–274, April 2007.
- [Sfakiotakis *et al.*, 2006] M. Sfakiotakis, D.P. Tsakiris, et A. Vlaikidis , « Biomimetic centering for undulatory robots », In *IEEE Int. Conf. on Biomedical Robotics and Biomechatronics, Pisa, Italy*, April 2006.
- [Smith, 1999] J.R. Smith , *Electric Field Imaging* , Thèse de Doctorat, Massachusetts Institute of Technology, 1999.
- [Solberg *et al.*, 2007] J.R. Solberg, K.M. Lynch, et M.A. MacIver , « Robotic electrolocation : Active underwater target localization with electric fields », In *IEEE, International Conference on Robotics and Automation, Rome, Italie*, pp. 4879–4886, 2007.

- [Susbielles et Bratuch, 1981] G. Susbielles et C. Bratuch , *Vagues et ouvrages pétroliers en mer* Editions Technip, 1981.
- [Triantafyllou et Triantafyllou, 1995] M.S. Triantafyllou et G.S. Triantafyllou , « An efficient swimming machine », *Scientific American*, vol. 272, pp. 64–70, Mars 1995.
- [Tytell et Lauder, 2004] E.D. Tytell et G.V. Lauder , « The hydrodynamics of eel swimming i. wake structure », *J. of Experimental Biology*, vol. 207, pp. 1825–1841, 2004.
- [Wolfgang *et al.*, 1999] M.J. Wolfgang, J.M. Anderson, M.A. Grosenbaugh, D.K.P. Yue, et M.S. Triantafyllou , « Near-body flow dynamics in swimming fish », *The Journal of Experimental Biology*, vol. 202, pp. 2303–2327, 1999.
- [Yamada *et al.*, 2005] H. Yamada, S. Chigisaki, M. Mori, K. Takita, K. Ogami, et S. Hirose , « Development of amphibious snake-like robot acm-r5 », In *The Proc. of 36th Int. Symposium on Robotics*, 2005.
- [Yu *et al.*, 2003] J. Yu, S. Wang, et M. Tan , « Basic motion control of a free-swimming biomimetic robot fish », In *IEEE Conference on Decision and Control, Maui, Hawaii USA*, pp. 1268–1273, December 2003.
- [Yu *et al.*, 2005] J. Yu, S. Wang, et M. Tan , « A simplified propulsive model of biomimetic robot fish and its realization », *Robotica*, vol. 23, pp. 101–107, 2005.

Annexe A

Expérimentation

A.1 Introduction

Dans le mémoire, nous avons étudié les différents aspects de la modélisation du robot anguille. Il est également important de pouvoir appliquer les résultats obtenus sur le prototype réel. Pour rappel, le prototype réalisé dans ce projet possède une structure hybride composée de 12 modules identiques dont chacun possède 3 moteurs à courant continu. Pour chacun de ces moteur, un asservissement devra être réalisé afin de reproduire les consignes articulaires qui seront calculées à partir du générateur de mouvements.

L'annexe est organisée de la manière suivante : Dans le paragraphe A.2, on commence par présenter l'architecture du prototype et le matériel utilisé, dans le paragraphe A.3 sont présentées les boucles de régulation des moteurs de l'anguille. Le paragraphe A.4 traite du système global de gestion et de commande de l'anguille et enfin le paragraphe A.5 présente l'architecture générale de l'électronique de commande.

A.2 Présentation du prototype

A.2.1 L'architecture informatique

Contrairement au cas de la simulation où tous les calculs sont effectués sur un seul ordinateur, le prototype est constitué d'un ensemble de plusieurs unités informatiques reliées en réseau. Ces unités informatiques sont classées selon deux types :

- Une unité informatique dans la tête de l'anguille qui joue le rôle de "cerveau".
- Six unités informatiques réparties le long de l'anguille et permettant chacune de contrôler deux modules consécutifs.

Ce type d'architecture est généralement plébiscité lors de la conception d'un prototype car elle permet de dissocier les éléments permettant le contrôle global du système (positionnement absolu, calcul des trajectoires, décision des mouvements,...) et le contrôle des tâches locales (pilotage des actionneurs, surveillance des capteurs,...) :

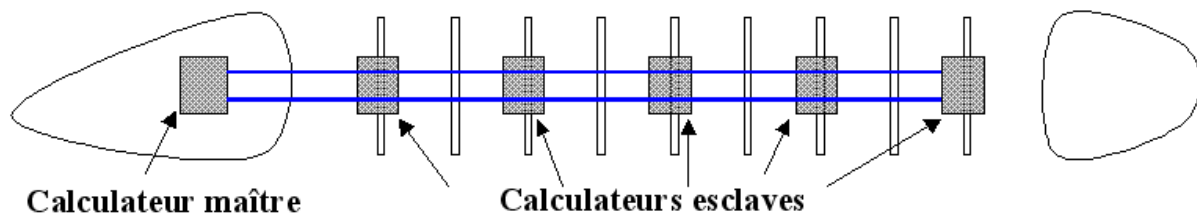


Figure A.1: – Répartition des unités informatiques dans le prototype

Dans le cas de l’anguille, cela permet de simplifier la programmation car elle permet d’avoir un seul programme commun à toutes les unités informatiques présentes dans les vertèbres, et un second programme dans l’unité informatique de tête qui n’a alors plus à se soucier du fonctionnement du corps.

Dans le cadre du projet robot anguille qui a pour objectif de créer un robot le plus bio-mimétique possible, ce type d’architecture est très importante car elle est en réalité directement inspirée du fonctionnement neuronal des animaux vertébrés. En effet, le cerveau n’a pas pour rôle de générer les mouvements locomoteurs mais tient le rôle de coordinateur. C’est au système nerveux de la moelle épinière à travers une multitude de réseaux de neurones que revient la tâche d’assurer le contrôle locomoteur de l’animal. Dans notre cadre d’étude, l’ensemble de ces réseaux de neurones locomoteurs sera remplacé par nos 6 unités informatiques et la moelle épinière sera substituée par un réseau de communication informatique.

A.2.2 Présentation du matériel utilisé

Nous avons vu que le prototype est piloté par 7 unités informatiques indépendantes. Ces unités sont des microcontrôleurs PowerPC MPC565 distribués par la société Phytec et sélectionnés pour les caractéristiques suivantes :

- unité de calcul flottant 64 bits ;
- capacité de générer au moins 6 PWM indépendantes ;
- capacité à gérer 6 codeurs en quadrature ;
- capacité à gérer au moins 6 convertisseurs Analogique/Numérique ;
- utilisation de 2 Bus CAN ;
- utilisation d’une liaison série ;
- vitesse de calcul suffisante pour le calcul des modèles géométriques inverses.

Tous les PowerPC sont équipés d’un mini système d’exploitation temps réel basé sur les spécifications OSEK-VDX, ce qui permet d’exécuter plusieurs tâches (ou programmes) en

parallèle. Comme précisé par les caractéristiques, la mise en réseau des microcontrôleurs se fait via 2 bus CAN (un principal + un de secours) qui permettront de faire transiter les ordres du cerveau de notre prototype vers les contrôleurs locaux.

Dans la première étape du prototype, les consignes de trajectoires seront envoyées par un opérateur via une radio-commande. Elles seront ensuite interprétées par le calculateur de la tête en consigne de nage pour être finalement envoyées par le réseau aux autres microcontrôleurs. Le robot anguille intègre également dans sa tête un contrôleur autonome qui a pour fonction la régulation de l'angle de roulis (rotation autour de l'axe de la tête) afin que le robot reste stable (principe similaire aux sous-marins) lors de la nage. Cette régulation est effectuée par l'intermédiaire d'une centrale inertielle connectée au PowerPC située dans la tête et capable de mesurer l'angle de roulis. Les deux autres angles donnés par la centrale sont quant à eux utilisés dans la commande générale de l'anguille pour gérer efficacement les changements de trajectoires demandés par l'opérateur. La tête de l'anguille est également équipée de deux capteurs de vitesse de type Loc (les systèmes à effet Doppler n'étant pas suffisamment précis pour de faibles vitesses) situés de part et d'autre de la tête et dont la moyenne permet de réguler la vitesse sur celle imposée par l'opérateur.

Le but des six autres PowerPC est de gérer l'ondulation des deux modules dont ils dépendent. Pour ces deux modules, le PowerPC doit gérer les boucles d'asservissement de 6 moteurs à courant continu 24V. Ceux-ci sont en réalité alimentés en 40V afin que leur couple maximum soit disponible à n'importe quelle vitesse de rotation (il faut sur-alimenter pour pouvoir contrer l'influence de la force contre électromotrice du moteur). Pour pouvoir piloter ces 6 moteurs, une carte de puissance permettant de commander les moteurs en couple est reliée au PowerPC (la carte est équipée d'un variateur de vitesse et d'une mesure de courant).

A.2.3 La méthodologie de commande adoptée

Nous avons vu jusqu'à présent comment le fonctionnement de l'anguille était distribué de manière bio-mimétique. Reste à savoir quel type de commande utiliser, et que va-t-on programmer dans les différentes unités informatiques.

Concernant la tête de l'anguille, il n'existe pas de commande unique mais une multitude de commandes suivant les objectifs et les critères que l'on se fixe. Des algorithmes de commandes ont été développés dans le cadre de multiples travaux [Yu *et al.*, 2003, Lapierre et Jouvencel, 2005, Alamir *et al.*, 2007, Ijspeert et Crespi, 2007], qui permettent au final de déterminer un jeu de coefficients utilisables pour la nage qu'on pourrait comparer aux

signaux qu'envoie le cortex cérébral à la moelle épinière de l'anguille biologique. Parmi ces coefficients, on trouve la fréquence d'oscillation, la longueur d'onde ou encore la courbure générale de l'anguille dans les différentes directions.

Pour les microcontrôleurs assurant le contrôle local de la locomotion, nous avons recensé 3 modes de fonctionnement possibles :

- un asservissement sur une (ou plusieurs) consigne(s) envoyée(s) par la tête à chaque pas de temps de calcul ;
- un asservissement sur une génération de mouvement sinusoïdale locale dont les paramètres sont modifiés par la tête lorsque cela est nécessaire ;
- un asservissement sur une génération de mouvements à base de réseaux de neurones (ou CPGs) s'auto-adaptant aux consignes de la tête.

Cette dernière possibilité, qui est la plus bio-mimétique n'est pas la méthode de commande retenue dans un premier temps car moins intuitive qu'une commande plus classique et plus complexe à mettre en oeuvre sur un système distribué en réseau (la communication devient complexe avec la multiplication des connexions synaptiques). C'est la deuxième solution que nous avons retenue car elle permet d'alléger le temps de calcul du PowerPC situé en tête (par rapport à la première méthode) et confère une autonomie locale, tout comme un système à base de réseaux de neurones, la complexité en moins. Le problème inhérent de ce type d'architecture et de commande est que l'ensemble des PowerPC doivent être à tout moment synchronisés si l'on veut que le fonctionnement global de tous les contrôleurs locaux soit cohérent. Ce genre de problème est tout de même facilement contournable si on intègre dans le réseau une horloge globale gérée par le calculateur maître.

A.3 Commandes bas niveau de chaque module

La tâche dédiée à la commande locale est un asservissement en position des différents moteurs du module afin de respecter la consigne calculée par le générateur de mouvements (fonction sinusoïdale). Cet asservissement est composé de deux régulateurs imbriqués : un régulateur de bas niveau de type boucle de courant qui permet de commander le moteur en couple, imbriqué dans un régulateur de type boucle de position.

Pour que ces deux boucles fonctionnent convenablement, il faudra respecter ces 3 conditions :

- la boucle de courant doit être vue comme un gain pur par la boucle de position ;
- la bande passante de la boucle de courant doit être largement supérieure à celle de la boucle de position ;

- la boucle de position doit avoir une fréquence d'échantillonnage plus faible que la boucle de courant.

A.3.1 Étude de la boucle de courant

La boucle de courant permet de réguler l'intensité sur une valeur de consigne et donc, modulo la constante de couple K_t , de réguler le couple du moteur. Pour réaliser cet asservissement, il existe deux types de régulateurs classiquement utilisés pour les boucles de courant :

- le régulateur PI (architecture parallèle),
- le régulateur I-P (architecture sériele).

Ces deux régulateurs sont quasiment identiques et lorsqu'on calcule la fonction de transfert en boucle fermée on obtient le même dénominateur. Seulement, le régulateur I-P possède l'avantage de ne pas introduire de zéro au numérateur de la fonction de transfert, ce qui est intéressant puisqu'on évite intrinsèquement de rajouter de l'instabilité et du bruit. La structure complète du régulateur est la suivante (figure (A.2)) :

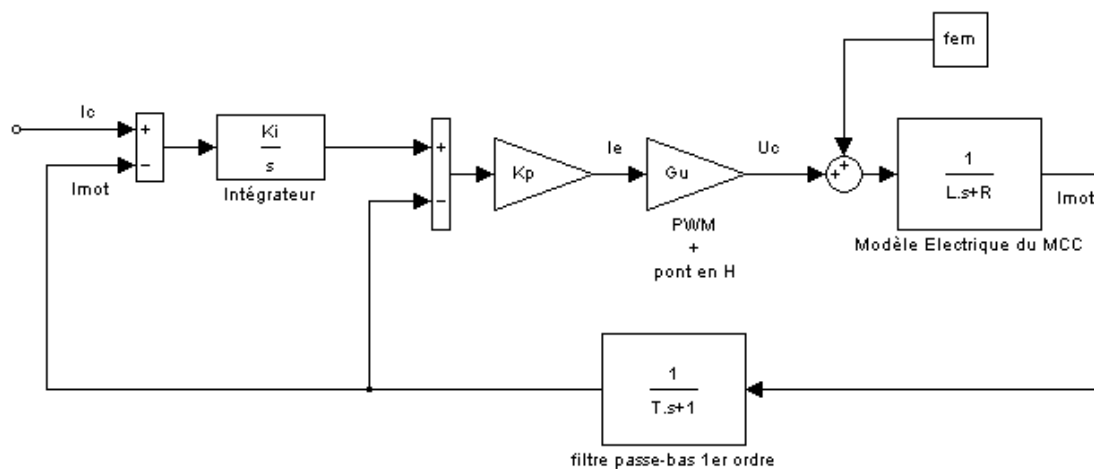


Figure A.2: – Structure du régulateur I-P pour la boucle de courant

Dans cette boucle de régulation, on retrouve les éléments suivants :

- K_i et K_p les coefficients du régulateur I-P ;
- G_u qui représente le gain entre le courant souhaité et la tension appliquée au moteur permettant de l'obtenir ;
- f_{em} qui représente la force contre électromotrice du moteur ;
- le modèle électrique du moteur ;
- un filtre passe bas du premier ordre qui permet de lisser le courant et de supprimer les effets de commutations du moteur introduits par le hachage de la tension d'alimentation.

Pour l'étude du régulateur, on se place dans le cas où le système est parfaitement continu et idéal (on ne hache pas la tension d'alimentation pour faire varier la tension moyenne appliquée au moteur). Le gain entre I_e et U_c s'exprime alors par :

$$G_u = R$$

R représentant la résistance de l'induit du moteur (voir caractéristiques en annexe B). Pour que le régulateur soit idéal on remplace dans un premier temps le filtre passe bas par un gain pur de 1.

La fonction de transfert s'écrit donc :

$$H(s) = \frac{I_c(s)}{I_m(s)} = \frac{K_i K_p G_u}{Ls^2 + (R + K_p G_u)s + K_i K_p G_u} \quad (\text{A.1})$$

On remarque que le gain statique de la boucle est bien de 1 comme précisé dans le cahier des charges. Cette fonction de transfert peut également être écrite sous la forme canonique suivante :

$$H(s) = \frac{K}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (\text{A.2})$$

On peut alors en déduire l'expression des gains K_p et K_i en fonction de ξ et ω_n , soit :

$$K_p = \frac{2\xi\omega_n L - R}{G_u} \quad (\text{A.3})$$

$$K_i = \frac{\omega_n^2 L}{K_p G_u} \quad (\text{A.4})$$

Ainsi, il est possible de régler les coefficients du régulateur en fonction de la bande passante souhaitée ω_n et du dépassement ξ . Si l'on souhaite obtenir la réponse la plus rapide et sans oscillations, on prend ($\xi = 1$) ce qui ramène le système à un second ordre à pôle réel double.

Dans ce cas, c'est donc le choix de ω_n qui va déterminer la valeur des coefficients du régulateur. Si l'on souhaite avoir le système le plus rapide possible, une méthode simple et efficace consiste à prendre pour valeur de ω_n la pulsation correspondant à la constante de temps électromécanique du moteur (qui représente la bande passante du moteur) :

$$\omega_n = 2\pi \frac{RJ}{K_t K_e}$$

avec K_t et K_e les constantes de couple et électrique du moteur et J l'inertie du rotor.

Le calcul de la constante électromécanique nous donne une bande passante d'environ 100Hz pour la boucle de courant.

A présent, il faut étudier l'influence de la présence du filtre et le caractériser pour que celui-ci respecte les différentes contraintes. La première contrainte est le respect de la fréquence de Shannon : on a choisi d'échantillonner la boucle de courant à 1kHz ce qui veut dire que la fréquence de coupure du filtre ne doit pas dépasser les 500Hz. La deuxième contrainte est que la fréquence de coupure du filtre doit se situer au-dessus de la bande passante de la boucle (100Hz), sinon on risque d'atténuer et de déphaser le signal ce qui aura pour conséquence de dégrader les performances. Finalement, on choisit de régler la fréquence de coupure du filtre à 300Hz. Pour apprécier les différences entre le régulateur sans filtre et le régulateur avec filtre, on commence par tracer leurs diagrammes de Bode (figure A.3) :

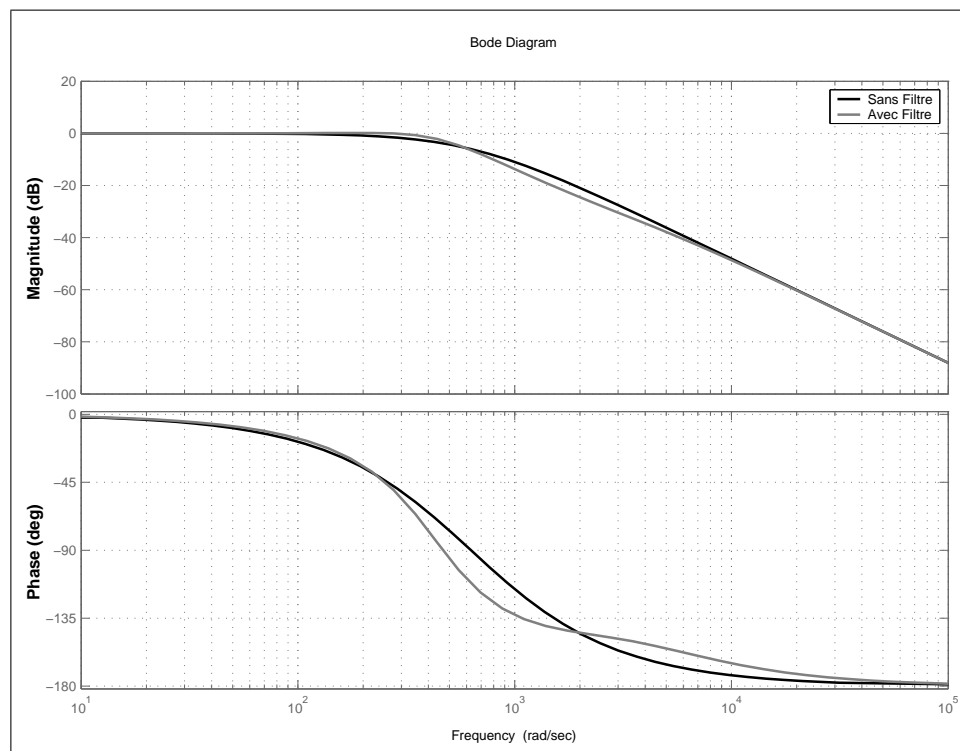


Figure A.3: – Diagramme de Bode pour le régulateur avec et sans filtre

Le diagramme de Bode nous montre que la présence du filtre dégrade légèrement les performances comme on pouvait s'y attendre. En effet, la présence du filtre augmente la rapidité du déphasage, mais il atténue moins le signal dans la bande passante du système. On remarque également que le gain de la boucle est légèrement supérieur à 1 sur la bande des 200-350 rd/s ce qui va sensiblement dégrader les performances du système. Cette

dégradation de performance provient d'une diminution de la marge de phase de la boucle. Pour quantifier cette diminution de la marge de phase, on trace le diagramme de black du régulateur en boucle ouverte (figure A.4) :

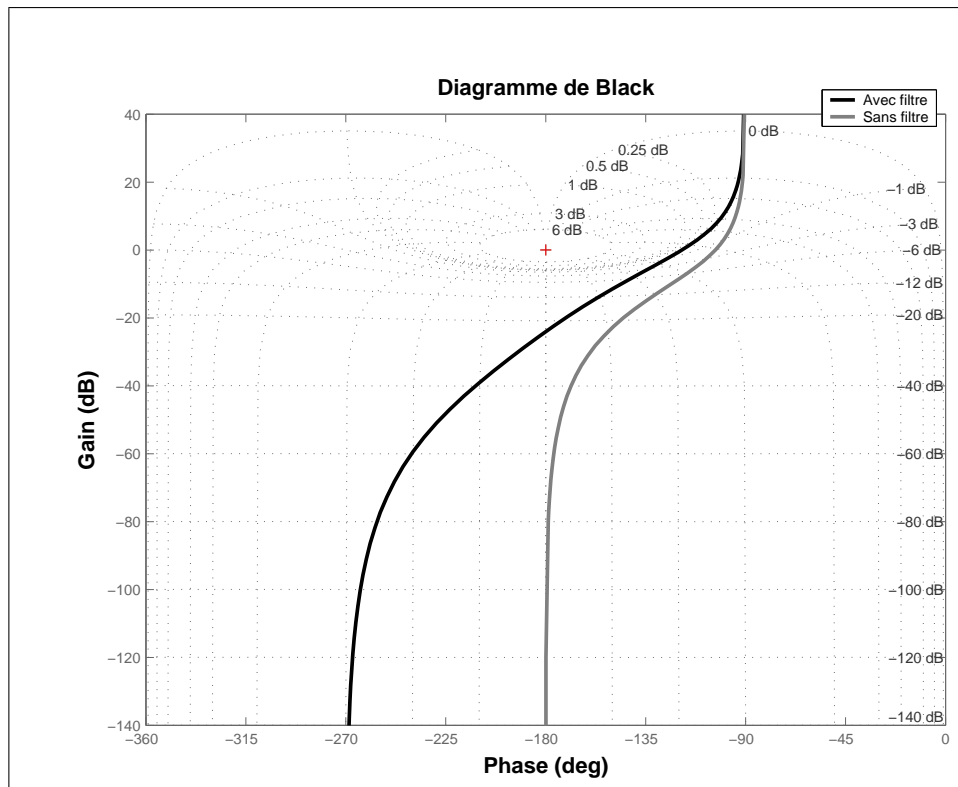


Figure A.4: – Diagramme de Black pour le régulateur avec et sans filtre

Sur la figure (A.4), on vérifie que la présence du filtre diminue la marge de phase. La mesure de ces marges pour les deux courbes tracées donne :

- une marge de phase de 76.3° pour le correcteur sans filtre ;
- une marge de phase de 61.3° pour le correcteur avec filtre.

En rajoutant le filtre dans la boucle de courant, nous n'avons pas rendu le système instable, et celui-ci reste suffisamment robuste. Cependant, le fait de diminuer la marge de phase va rendre le système un peu plus oscillant. On peut vérifier ce résultat si on trace la réponse à un échelon du régulateur avec et sans filtre :

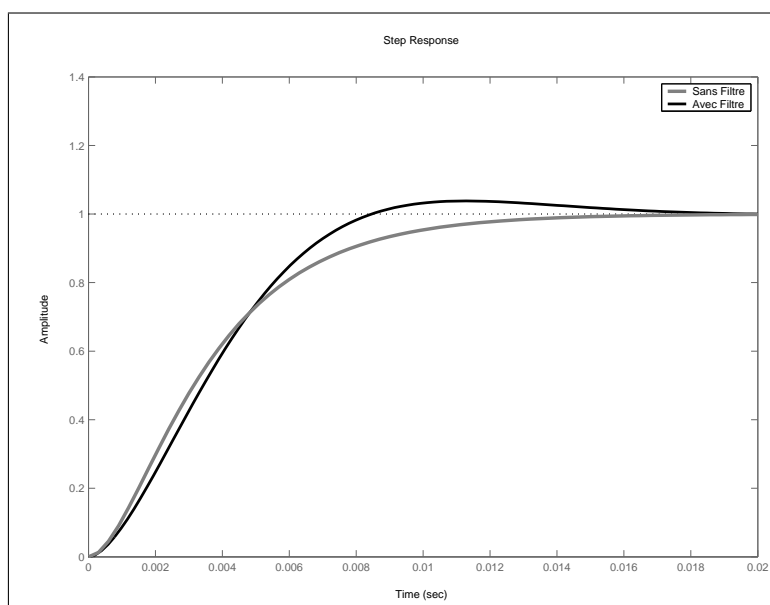


Figure A.5: – Réponse à un échelon du régulateur avec et sans filtre

Le régulateur sans filtre étant réglé avec un $\xi = 1$, on observe que la réponse à un échelon ne présente pas de dépassement de la consigne comme souhaité. Avec le filtre, comme observé sur les diagrammes de Bode et de Black, le système devient légèrement oscillant et provoque un dépassement de la consigne. La présence du filtre (qui est obligatoire) impose donc un réglage manuel de ξ si on souhaite avoir une réponse optimale. Il est également possible d'améliorer sensiblement le résultat en augmentant la fréquence de coupure du filtre (tout en restant en-dessous des 500 Hz). L'augmentation de la fréquence de coupure a pour effet de décaler l'influence du filtre sur le diagramme de Bode, d'augmenter la marge de phase et par conséquent de diminuer le caractère oscillant. Finalement, pour le prototype, nous avons décidé de garder la fréquence de coupure de 300Hz : les résultats de simulations montrent qu'il faut alors prendre $\xi = 1,65$ pour ne pas avoir de dépassement (pour une bande passante de 100Hz et une fréquence d'échantillonnage de 1kHz).

A présent que l'étude en continue est terminée, il faut penser à traiter le cas des non linéarités qui peuvent apparaître dans ce genre de régulateur. Les non linéarités qu'il faut prendre en compte dans ce régulateur sont les limites des actionneurs. Dans notre cas, il s'agit de la limite en tension qu'on peut appliquer au moteur. Le schéma utilisé sera alors celui de la figure (A.6) :

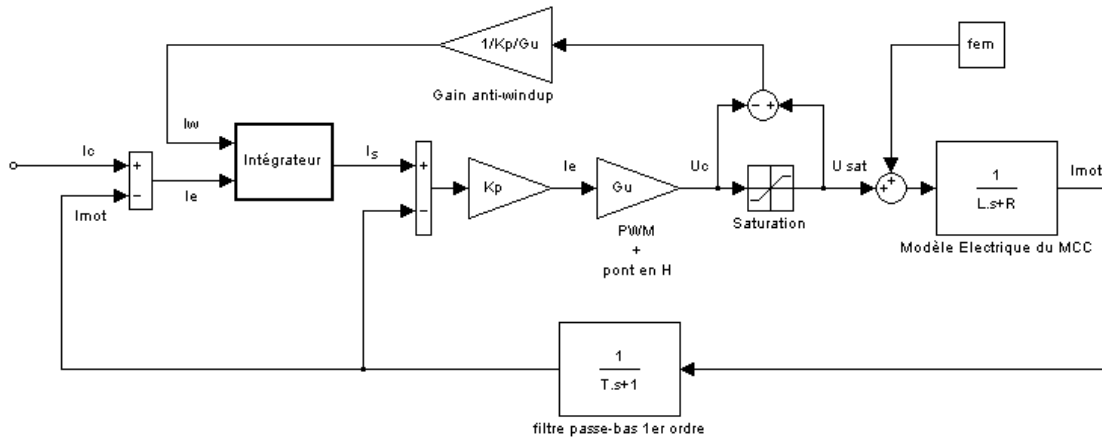


Figure A.6: – Structure du régulateur I-P pour la boucle de courant

On peut observer sur ce schéma la présence de la saturation de commande de la tension du moteur. Sur ce schéma, on a également ajouté sur l'entrée de l'intégrateur un terme d'anti-windup I_w (anti-emballement en français) qui permet d'éviter au terme intégral de la boucle de diverger lorsque la commande arrive en saturation. On peut observer que ce dispositif est inopérant lorsqu'il n'y a pas de saturation ($U_c(t) \leq U_{Sat}$). Lorsque le système est en saturation, le terme d'anti-windup va faire en sorte de faire tendre la commande $U_c(t)$ vers la tension de saturation U_{Sat} . L'expression de l'intégrateur permettant de prendre en compte l'anti-windup s'exprime par la fonction de transfert suivante :

$$I_S = K_I \frac{T_e}{2} \frac{1 + z^{-1}}{1 - z^{-1}} I_e + \frac{z^{-1}}{1 - z^{-1}} I_w \quad (\text{A.5})$$

avec $T_e=1$ ms la période d'échantillonnage de la boucle de courant (pour une fréquence de 1kHz).

Cette équation nous montre que l'action de l'anti-windup ne vient pas modifier le signal d'entrée de l'intégrateur mais vient directement modifier sa sortie. Ce principe permet d'éviter de modifier le signal I_e et permet également d'améliorer la vitesse de réponse du régulateur lorsque celui-ci sort de la zone de saturation.

La figure (A.7) présente le gain de performance qu'apporte l'anti-windup pour notre régulateur lorsque celui-ci entre en saturation :

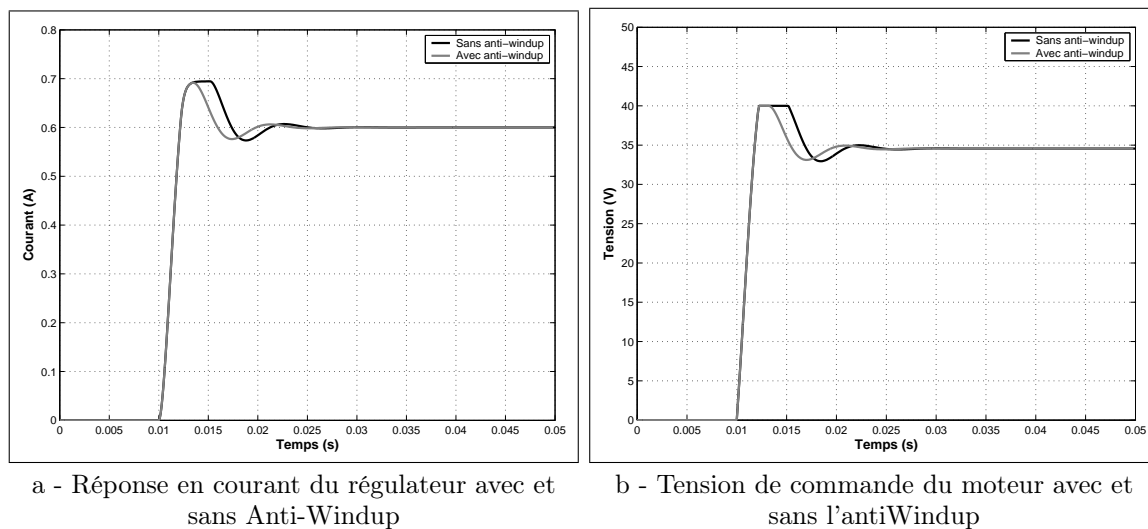


Figure A.7: – Influence de l'anti-windup sur les performances du régulateur en cas de saturation

Pour un réglage du régulateur avec un $\xi = 0.707$, on remarque que sans l'anti-windup le courant reste saturé pendant quelques instants après le premier dépassement. Cette saturation correspond au temps qu'il faut à l'intégrateur pour se décharger. A l'inverse, la présence de l'anti-windup permet d'avoir une réponse sans saturation du courant et un temps de réponse plus rapide. On remarque également que l'anti-windup permet de diminuer la durée de la saturation en tension du moteur d'où le gain de performance.

Comme pour tout moteur, la perturbation due à la force contre-electromotrice est extrêmement importante à prendre en compte, puisqu'en fonction de la vitesse de rotation du moteur, il est tout à fait possible d'annuler la tension de commande du moteur et par conséquent de rendre impossible la régulation de courant. La figure (A.8) montre les caractéristiques classiques d'un moteur à courant continu :

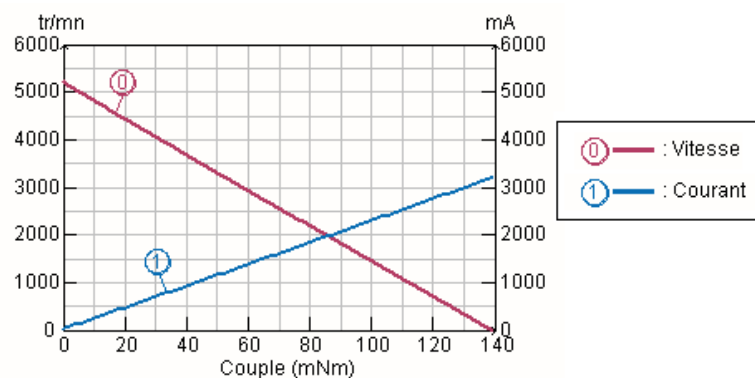


Figure A.8: – Caractéristiques d'un moteur à courant continu

Cette figure nous montre qu'à vitesse maximum, le moteur ne peut plus fournir de couple ce qui est gênant pour notre régulateur. Pour palier à ce problème, la solution consiste à suralimenter suffisamment les moteurs pour décaler la courbe de couple vers le haut. Il faudra cependant faire attention de ne pas dépasser la vitesse maximum du moteur pour ne pas le détériorer. Les calculs montrent que pour avoir à notre disposition les 2N.m sur la plage de vitesse qui nous intéresse, il faut passer d'une tension d'alimentation de 24V à une tension d'alimentation de 40V.

A.3.2 Etude de la boucle de position

La boucle de position permet de faire une régulation de la position du moteur vis-à-vis des consignes demandées, mais permet également de calculer le couple (donc l'intensité) de consigne que devra réguler la boucle de courant. Pour étudier ce régulateur, on se place dans le cas où le moteur est considéré comme idéal (les frottements secs et visqueux sont simplement considérés comme des perturbations extérieures). Le modèle dynamique inverse d'un moteur sans frottement s'écrit :

$$\Gamma = J \ddot{q} \quad (\text{A.6})$$

où J représente l'inertie du moteur et \ddot{q} son accélération angulaire.

On utilise ici un régulateur PID série ce qui nous permet de ne pas introduire de zéro au numérateur de la fonction de transfert. La boucle de position (sans consigne de vitesse ni d'accélération) se schématise de la manière suivante :

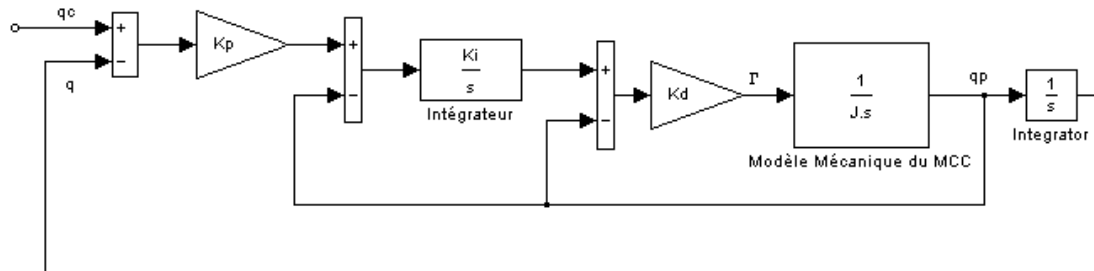


Figure A.9: – Structure du régulateur PID série pour la boucle de position

Malgré le fait que le régulateur ne prenne ici que des consignes en position, on observe que celui-ci utilise un retour tachymétrique pour effectuer sa régulation. L'avantage du retour tachymétrique est qu'il permet d'augmenter la stabilité du système et donc de choisir une plus grande plage de valeurs pour les gains du régulateur. Le transfert de ce régulateur s'exprime de la manière suivante :

$$\frac{q}{q_d} = \frac{K_p K_i K_d}{J s^3 + K_d s^2 + K_d K_i s + K_p K_i K_d} \quad (\text{A.7})$$

Comme pour la boucle de courant, il est possible de réécrire cette fonction de transfert sous forme canonique et factorisée sous la forme :

$$\frac{q}{q_d} = \frac{K}{(s + \omega_n)(s^2 + 2\xi\omega_n s + \omega_n^2)} \quad (\text{A.8})$$

Par identification, on peut alors déterminer les gains du régulateur en fonction de ξ et ω_n :

$K = K_p K_i K_d$	$K_d = J(2\xi + 1)\omega_n$	$K_i = \omega_n$	$K_d = \frac{\omega_n}{2\xi + 1}$
-------------------	-----------------------------	------------------	-----------------------------------

On observe alors que le gain statique de cette fonction de transfert est de 1 et qu'elle est toujours stable quelles que soient les valeurs de ξ et ω_n . Cependant, contrairement à la boucle de courant où le choix de ω_n est relativement libre, le réglage de la boucle de position est plus difficile. En effet, pour que la boucle de courant soit transparente vis-à-vis de la boucle de position, il est nécessaire d'avoir une fréquence d'échantillonnage plus faible. En général, on considère qu'un facteur 10 est suffisant, ce qui nous amène dans notre cas à échantillonner la boucle de position à 100Hz. Il est possible de descendre en dessous des 100hz pour l'échantillonnage, compte tenu des faibles dynamiques qu'auront les mouvements de l'anguille, mais on dégrade d'autant la précision du calcul de la vitesse du moteur (la vitesse est obtenue par dérivation numérique du signal des codeurs).

Jusqu'à présent pour le régulateur de position, on n'a pas considéré les non-linéarités du moteur ainsi que le problème des saturations. Le moteur électrique n'étant pas idéal, il est nécessaire de lui rajouter les frottements secs et visqueux ainsi que la saturation correspondant au couple maximum qu'il est possible de délivrer. Le schéma de la boucle de position modifié est représenté sur la figure (A.10) :

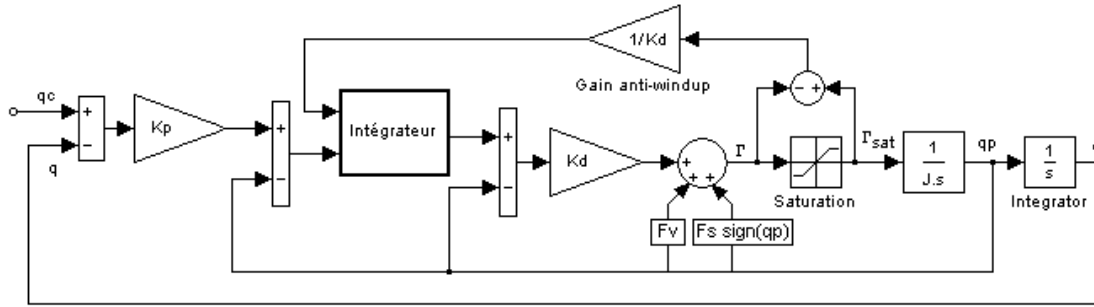


Figure A.10: – Boucle de position tenant compte des saturations et des non-linéarités

Dans le cas où la boucle de position est couplée avec une boucle de courant, l'introduction du terme des frottements secs est important, surtout si l'application nécessite régulièrement des changements de sens de rotation (ce qui est le cas du robot anguille). En effet les changements de sens répétitifs font que le terme intégral du régulateur doit constamment réguler le frottement sec comme une perturbation. Si le frottement sec est compensé dans le calcul du couple, la performance du régulateur est fortement améliorée au niveau du temps de réponse du système. Si on souhaite augmenter un peu plus la rapidité du système, une astuce consiste à ne plus calculer F_s en fonction du signe de la vitesse mais en fonction du signe de l'erreur entre la consigne et la mesure de la position. Cette astuce est efficace lorsque l'erreur change de signe alors que le moteur tourne toujours dans le même sens : dans le cas classique, prendre le signe de la vitesse fait que le frottement sec est un frein important dans la commande tant que le moteur ne s'est pas arrêté, tandis qu'en prenant le signe de l'erreur, on augmente la réactivité de la commande permettant ainsi de freiner le moteur plus rapidement pour le faire changer de sens.

Comme pour la boucle de courant, on introduit un terme d'anti-windup qui permet d'augmenter la rapidité de la réponse et d'éviter les instabilités du régulateur en cas de saturation de celui-ci. Le calcul de l'intégrateur se fait de la même manière que pour la boucle de courant en utilisant l'équation (A.5) avec pour T_e la période d'échantillonnage de la boucle de position.

Pour pouvoir lier le couple calculé par la boucle de position et le courant de consigne attendu pour la boucle de courant, on utilisera la relation bien connue :

$$\Gamma = K_T I \quad (\text{A.9})$$

A.3.3 Résultats expérimentaux

A partir des études théoriques pour les boucles de courant et de position, nous avons programmé dans le système informatique embarqué les deux régulateurs proposés. Nous avons ensuite réglé les deux régulateurs avec les coefficients suivants :

- pour la boucle de courant : $\omega_n = 2\pi.100$; $\xi = 1,65$ (bande passante de 100Hz) ;
- pour la boucle de position : $\omega_n = 2\pi.2$; $\xi = 1$ (bande passante de 2Hz).

Boucle de courant

La boucle de courant étant la boucle de plus bas niveau, on commence par tester si son fonctionnement est convenable et correspond à l'étude théorique menée précédemment. Sur la figure (A.11), on trace la réponse en courant du moteur pour une consigne de courant de $\pm 0,02A$:

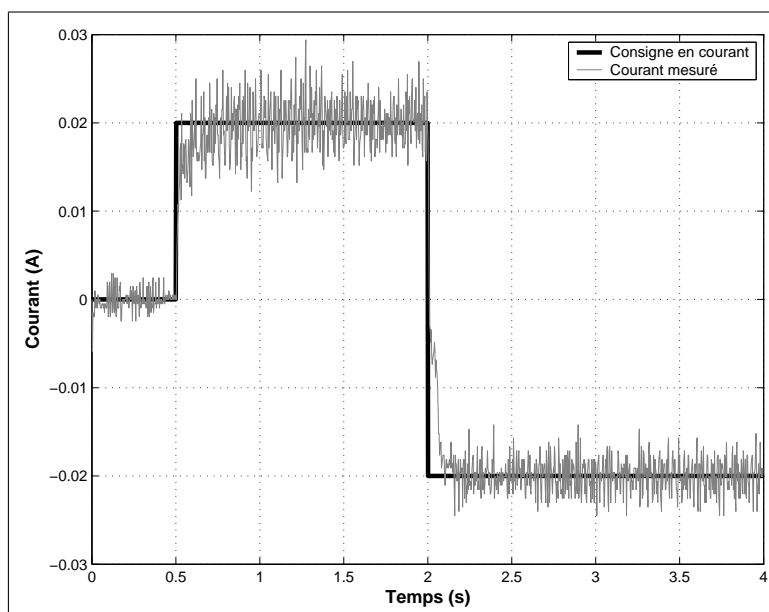


Figure A.11: – Test de la boucle de courant

La figure (A.11) nous montre que même avec le filtre de 300Hz utilisé, le signal du courant reste assez bruité. Cependant, on peut observer que la réponse à un échelon ne présente pas de dépassement de la consigne, ce qui est en accord avec l'étude théorique. On remarque également que le temps de réponse est très rapide au moment du changement de consigne (la pente est presque verticale) mais que la courbe est rapidement perturbée et ralentie ce qui provoque un temps de réponse de l'ordre de 0,15 s. Cette perturbation provient de la force contre-électromotrice du moteur qui vient diminuer le courant réellement fourni au moteur. Comme la partie mécanique du moteur possède une constante de

temps plus faible que le courant, l'effet de la fem (qui est proportionnelle à la vitesse de rotation du moteur) vient perturber la boucle de courant de manière retardée et allonge son temps de réponse.

Boucle de position

En amont de la boucle de courant que l'on vient d'expérimenter, on place le régulateur de position. Sur les figures (A.12 -a) et (A.12 -b), on trace la réponse en position du moteur pour une consigne en échelon de $\pm 10^\circ$ et $\pm 90^\circ$:

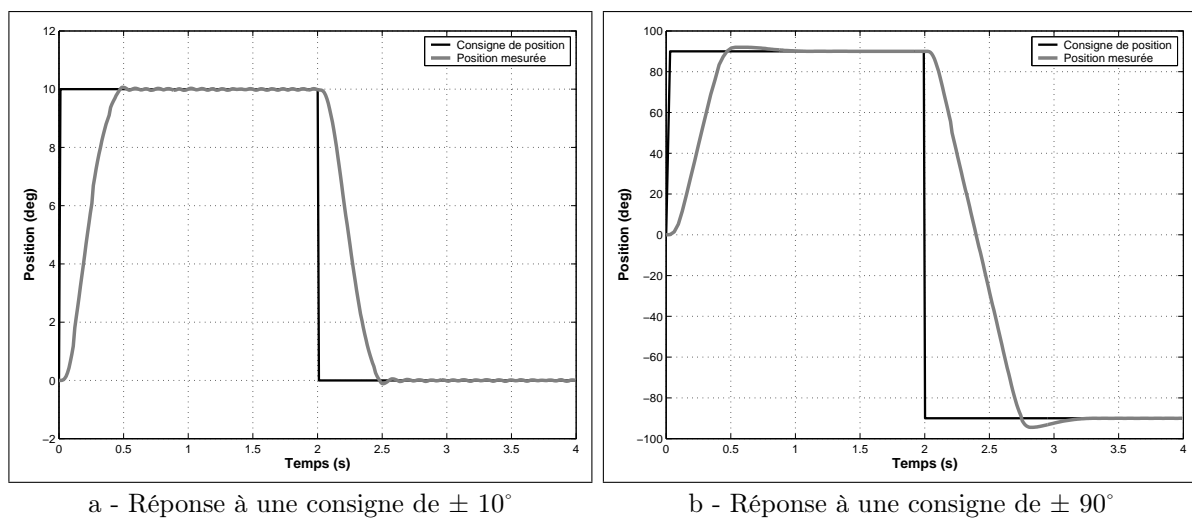


Figure A.12: – Réponse de la boucle de position pour différentes valeurs de consigne

On remarque sur la figure (A.12-a) que la position du moteur converge sans problème vers la consigne spécifiée et sans effectuer de dépassement comme on pouvait s'y attendre ($\xi = 1$). Sur la figure (A.12-b), on s'aperçoit que pour des consignes plus grandes, la réponse présente un petit dépassement. Ceci provient du fait qu'on arrive en saturation de la commande et qu'il n'est pas possible d'obtenir les performances optimales. Cependant, le dépassement est extrêmement léger grâce au terme d'anti-windup et reste, par conséquent, tout à fait acceptable.

Sur la figure (A.12-a), on peut noter qu'une fois la valeur de consigne atteinte, la courbe se met à osciller légèrement. Ces oscillations traduisent ici le problème du frottement sec qui intervient lors du changement de sens de rotation du moteur. Pour la réponse à une consigne de $\pm 10^\circ$, on peut alors tracer la courbe de la consigne de courant calculée par la boucle de position (voir figure A.13) pour mieux observer le phénomène.

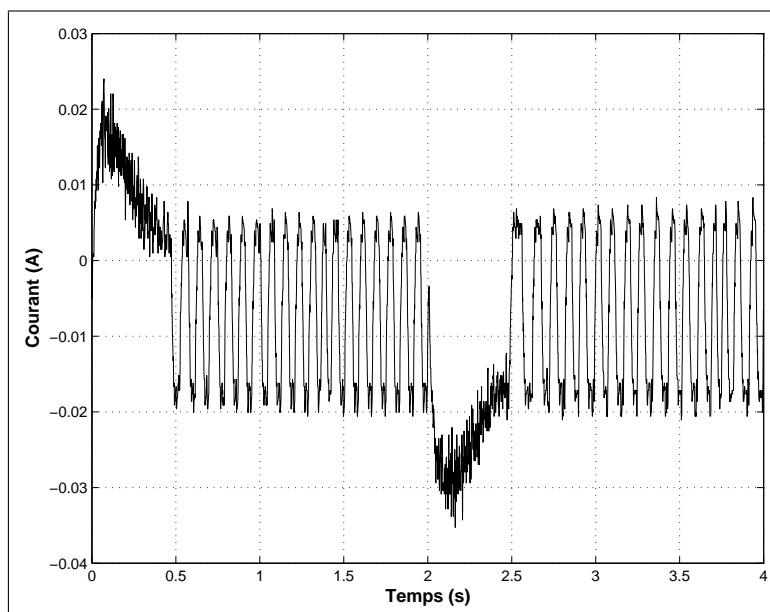


Figure A.13: – Consigne de courant calculée par la boucle de position

On retrouve sur la figure (A.13) le caractère oscillant de la courbe de position. On remarque également que les amplitudes des oscillations sont toujours les mêmes lorsque le moteur a atteint la position désirée. On met donc en évidence ici la courbe en hystérésis du frottement sec qui induit la présence d'un couple de démarrage non nul pour faire bouger le moteur. Même si on a tenu compte de ce frottement dans la commande, il s'agit ici d'une valeur théorique : une étape d'identification permettrait donc d'obtenir un meilleur résultat.

A.4 Commandes haut niveau de chaque module

Pour la commande de haut niveau (c'est-à-dire la génération de mouvements, le calcul des modèles géométriques, les gestions d'erreurs, etc), le système que nous avons étant distribué sur plusieurs unités informatiques, un procédé permettant l'interaction est nécessaire. Dans [Mihalachi, 2000], l'auteur propose d'implémenter dans chaque unité informatique un automate de contrôle qui permet d'exécuter différents programmes (ou tâches) selon la situation où l'on se trouve. A partir de ses travaux, nous avons adapté l'automate de contrôle pour les besoins de notre robot :

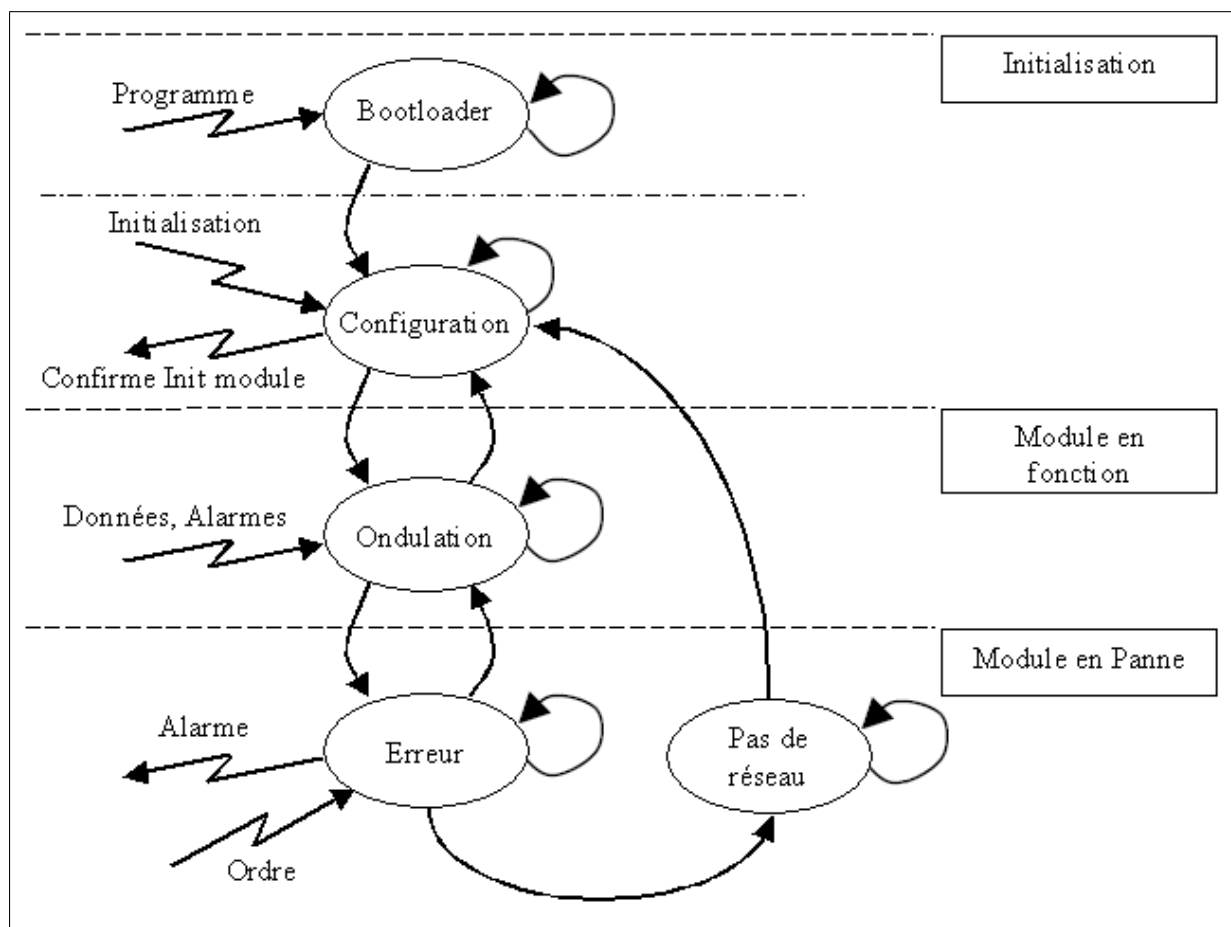


Figure A.14: – Automate de contrôle pour les unités informatiques situées dans les vertèbres de l'anguille

Cet automate est divisé en trois modes de fonctionnement principaux : le mode d'initialisation, le mode module en fonction et le mode module en panne. Pour chacun d'entre eux, les rôles sont :

1. Mode d'initialisation : l'initialisation d'une unité informatique est séparée en deux étapes : le bootloader et la configuration.
 - Le bootloader est un petit bout de programme qui permet de télécharger puis d'exécuter le logiciel principal. Ce procédé est extrêmement intéressant, puisqu'il évite d'avoir à reprogrammer l'unité informatique et donc de démonter, puis remonter l'ensemble de la mécanique et de l'électronique (cela permet d'éviter des pertes d'étanchéité et de fragiliser les différents composants) ;
 - La configuration constitue l'étape initiale du logiciel qui vient d'être téléchargé. Elle permet de configurer d'une part les différentes entrées-sorties utilisées pour piloter les moteurs, et d'autre part de configurer la communication réseau (où se

trouve l'unité informatique dans l'anguille par exemple).

2. Mode module en fonction : c'est dans ce mode que l'anguille va faire onduler son corps et que chaque module va devoir effectuer des mouvements sinusoïdaux. Pour que la synchronisation entre les différentes unités informatiques soit opérante et que les paramètres de la nage soit connus, la tâche liée à ce mode va recevoir les données nécessaires par le réseau.
3. Mode module en panne : Lorsqu'une panne est détectée, elle est alors signalée sur le réseau afin que les autres unités informatiques puissent en tenir compte. Si jamais le problème est une erreur de réseau, alors le programme se réinitialise.

Ce genre d'architecture est relativement simple à mettre en place avec un système d'exploitation temps réel puisque chaque mode de fonctionnement va consister en une ou plusieurs tâches qui pourront être exécutées et interrompues à tout moment selon le contexte de travail. De même il est possible d'attribuer des priorités plus ou moins importantes selon le rôle des tâches. Pour illustrer, on prend l'exemple des différentes tâches nécessaires pour générer l'ondulation des modules (figure A.15) :

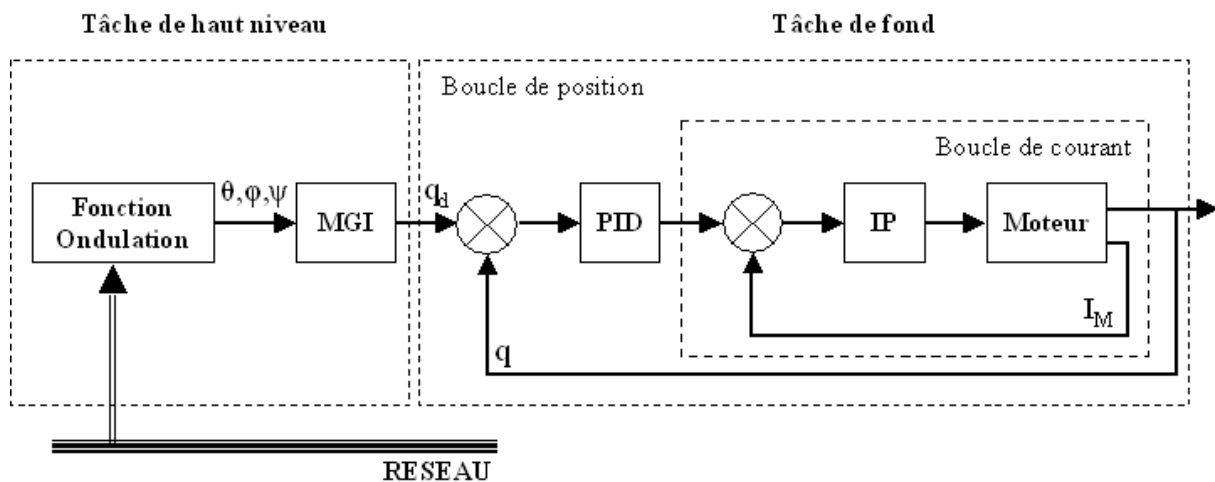


Figure A.15: – Schéma global de fonctionnement de la commande pour un module de l'anguille

On remarque sur la figure (A.15) que l'ensemble du schéma de commande des moteurs de l'anguille ne s'effectue pas sur une seule tâche mais sur deux. En effet, on a vu que les boucles de régulation doivent se faire sur un intervalle de temps très court et qui ne varie pas. Pour cela, les boucles sont programmées dans une tâche dite de fond, c'est-à-dire qu'elle s'exécute de manière indépendante du reste et n'est jamais interrompue. Pour le

calcul des consignes à envoyer aux moteurs, on crée une nouvelle tâche qui pourra avoir une période d'activation plus au moins longue suivant la fréquence de calcul que l'on souhaite avoir. Pour cette tâche, plusieurs options peuvent être mises en place selon le principe de fonctionnement de l'anguille :

- si la fonction d'ondulation est autonome (c'est-à-dire qu'elle n'est pas synchronisée avec les autres unités informatiques), la tâche sera activée de manière périodique et viendra prendre de manière asynchrone les paramètres de la nage dans un tableau de valeurs mis à jour par le réseau ;
- si la fonction d'ondulation est synchronisée, la tête de l'anguille enverra un signal de synchronisation (avec les paramètres de la nage) et la tâche sera exécutée à la réception de celui-ci. Ce mode de fonctionnement sera privilégié car il permet d'avoir un fonctionnement parfaitement synchrone de toute l'anguille et d'avoir une gestion du temps qui dépend uniquement de l'unité informatique dans la tête de l'anguille.

A.5 Electronique de commande des moteurs

Pour pouvoir actionner les moteurs du prototype, nous avons besoin d'une interface de puissance entre le microcontrôleur et le moteur. Dans le cahier des charges pour la commande des moteurs, l'interface de puissance développée doit pouvoir faire varier la vitesse du moteur dans les deux sens de rotation et être capable de mesurer le courant absorbé par le moteur. Sur ces spécifications, le schéma général suivant a été adopté (figure A.16) :

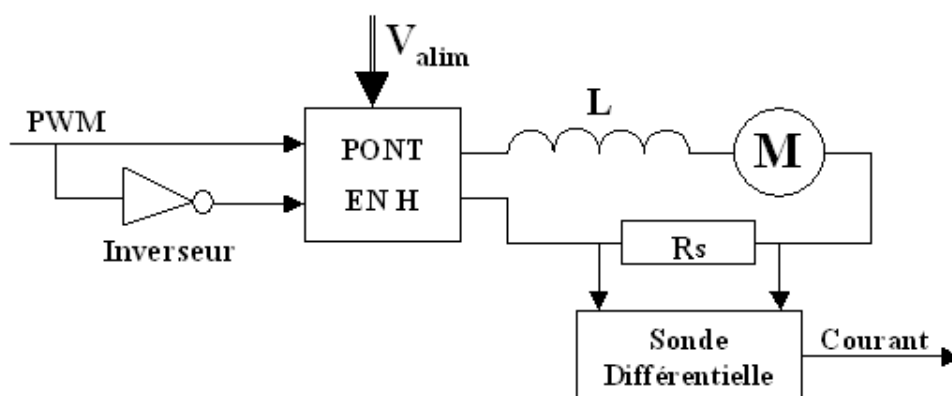


Figure A.16: – Schéma électronique de commande d'un moteur

Le schéma de la figure (A.16) est constitué de plusieurs parties :

- un signal PWM (modulation à largeur d'impulsion en français) et un inverseur pour faire varier la vitesse du moteur et permettre le changement de sens de rotation ;

- un pont en H : Ce composant permet de faire l'interface entre les signaux logiques de commande et la puissance délivrée au moteur ;
- une self L ;
- le moteur M ;
- une résistance de shunt R_s pour la mesure du courant ;
- une sonde différentielle pour mesurer et mettre en forme le courant moteur.

Le principe de commande d'un moteur par l'intermédiaire d'un signal et de son complément (l'inverseur sur le schéma) permet de n'avoir qu'un seul signal pour faire varier la vitesse et changer les sens de rotation. Ainsi, pour un rapport cyclique de 50%, le moteur sera à l'arrêt et pour 0% et 100% il tournera à plein régime dans l'un ou l'autre sens. La variation du rapport cyclique permet donc de moduler la tension moyenne du moteur et par conséquent le courant moyen. La figure (A.17) montre le principe d'alimentation alternée du moteur, et la figure (A.18) représente le courant moteur obtenu :

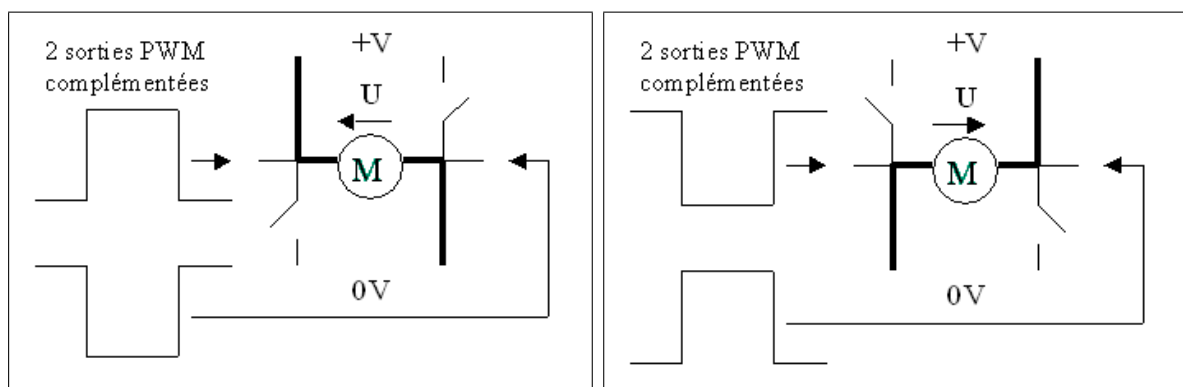


Figure A.17: – Alimentation du moteur suivant les phases du signal PWM

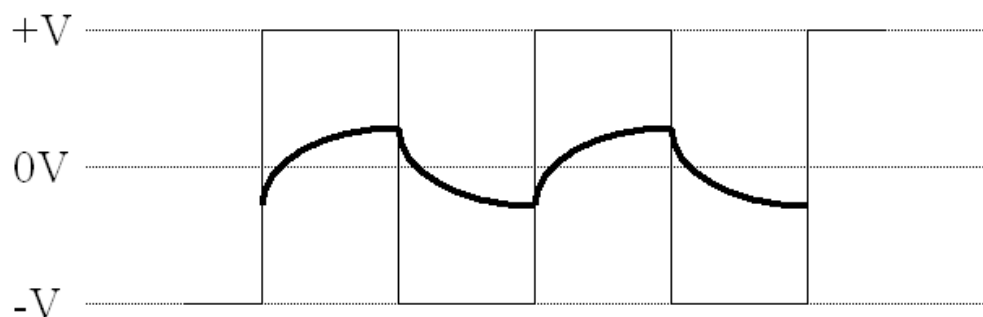


Figure A.18: – Courbe de courant de l'induit du moteur en fonction de la tension appliquée au moteur

On constate que pour un rapport cyclique de 50%, les valeurs moyennes de la tension et

du courant sont nulles, mais que la valeur efficace du courant ne l'est pas. En effet, même si le moteur ne tourne pas, le courant effectue des ondulations et induit une puissance électrique sous forme d'effet Joule (à cause de la résistance de l'induit du moteur). Le moteur n'étant pas ventilé à l'arrêt, si la puissance dissipée est trop importante, on risque alors de l'endommager. Pour éviter ce genre de problème, on considère généralement que la variation du courant crête-crête ne doit pas dépasser 10% du courant nominal du moteur. Par conséquent, pour le moteur à l'arrêt, le courant doit être compris dans l'intervalle $[-0,05 I_{\text{Nom}}; +0,05 I_{\text{Nom}}]$.

Le courant de l'induit du moteur (lorsqu'il ne tourne pas) est solution de l'équation différentielle suivante :

$$U = L \frac{di}{dt} + Ri \quad (\text{A.10})$$

dont la solution s'exprime par :

$$i(t) = \left(\frac{U}{R} - i_0 \right) \left(1 - e^{-\frac{t}{\tau}} \right) + i_0 \quad (\text{A.11})$$

avec :

- i_0 le courant initial ;
- τ la constante de temps égale à $\frac{L}{R}$.

En considérant que la période du signal PWM est très petite devant la constante de temps électrique, on peut alors approximer le début de l'exponentielle par une droite dont la pente correspond à la tangente à l'origine du signal $i(t)$. Ainsi, on peut considérer que l'équation temporelle du courant devient :

$$i(t) = \frac{U - Ri_0}{L} t + i_0 \quad (\text{A.12})$$

Si à présent on se place dans le cas où, pour $t=0$, $i_0 = -0,05 I_{\text{Nom}}$, alors pour $t=T/2$ (avec T la période du signal PWM) le courant ne devra pas dépasser $0,05 I_{\text{Nom}}$. Cela nous amène à la condition suivante :

$$T \leq \frac{0,2 L I_{\text{Nom}}}{U + 0,05 R I_{\text{Nom}}} \quad \text{ou} \quad L \geq \frac{T(U + 0,05 R I_{\text{Nom}})}{0,2 I_{\text{Nom}}} \quad (\text{A.13})$$

L'une ou l'autre de ces conditions nous assure d'être dans une plage de fonctionnement qui ne détériore pas le moteur. Pour savoir laquelle de ces deux conditions utiliser, on fait le calcul numérique pour les deux types de moteurs utilisés (voir annexe B) :

- pour le moteur de la jambe 3 d'un module, nous avons :

$$L=1,44 \text{ mH} ; I_{\text{Nom}}=179 \text{ mA} ; R=57,6 \Omega ; U=40 \text{ V}$$

L'application numérique de l'équation (A.13) nous donne : $f= 1/T < 785 \text{ kHz}$.

- pour les moteurs des jambes 1 et 2, nous avons :

$$L=0,8 \text{ mH} ; I_{\text{Nom}}=172 \text{ mA} ; R=36,3 \Omega ; U=40 \text{ V}$$

L'application numérique de l'équation (A.13) nous donne : $f= 1/T < 1.47 \text{ MHz}$.

Pour les deux types de moteurs, la fréquence maximale admissible du signal PWM est beaucoup trop grande (785 kHz) et ne peut pas être réalisée par les unités informatiques. Compte tenu des possibilités, nous avons choisi d'utiliser une fréquence de 78 khz. Avec cette nouvelle donnée, on calcule la deuxième condition de l'équation (A.13) permettant de déterminer la valeur minimale de la self. L'application numérique de ce calcul donne :

- pour le moteur de la jambe 3 :

$$L > 14 \text{ mH}$$

- pour les moteurs des jambes 1 et 2 :

$$L > 15 \text{ mH}$$

Ces résultats montrent qu'il est nécessaire d'ajouter une self en série du moteur afin de limiter les pertes par effet Joule dans celui-ci. La valeur de self à ajouter étant importante, nous avons fait en sorte de choisir une fréquence du signal PWM le plus haut possible afin que la self soit la plus petite possible (en valeur et en encombrement).

A.6 Conclusion

Dans cette annexe, nous avons présenté les différents développements menés pour le prototype. Nous avons présenté le matériel utilisé et les contraintes technologiques qu'il doit respecter. L'architecture que nous avons adoptée pour le prototype, est une architecture multi-processeurs qui permet d'augmenter, et de mieux répartir, la puissance de calcul pour la commande des modules de l'anguille.

Nous avons pu mettre en place des boucles d'asservissement en courant et en position, avec des régulateurs robustes et faciles à régler suivant les performances souhaitées. Nous avons fait le choix ici de ne pas faire une commande dynamique complète car d'une part, le calcul des modèles cinématiques impose des temps de calculs plus longs, et d'autre part, les mouvements qui seront donnés aux moteurs étant assez lents, ce type de commande ne représente pas un gain de performance significatif.

Le développement de l'architecture informatique n'est pas complètement définie mais les grandes lignes ont été présentées dans cette annexe. L'utilisation d'un système d'exploitation, même s'il n'est pas absolument nécessaire, permet de grandement simplifier la mise en place des différents programmes. De plus, la mise en place d'un automate de contrôle permet de créer un organigramme de comportement qui permet à l'utilisateur de mieux caractériser le système.

Enfin, à partir des caractéristiques des moteurs du prototype (voir annexe B), nous avons pu dimensionner l'électronique de commande afin que celle-ci fonctionne dans des conditions optimales.

Annexe B

Caractéristiques des moteurs de l'anguille

Chaque module de l'anguille que nous avons étudié dans ce mémoire embarque 3 moteurs (de la marque faulhaber). Ces moteurs sont répartis en deux groupes :

- deux moteurs identiques de la série 2224 pour les jambes 1 et 2. Ce sont des moteurs 24V équipés d'un réducteur de 1 :391 ;
- un moteur de la série 1727 pour la jambe 3. C'est un moteur 24V équipé d'un réducteur de 1 :246.

Le placement de ces moteurs sur le module est représenté sur la figure (B.1) :

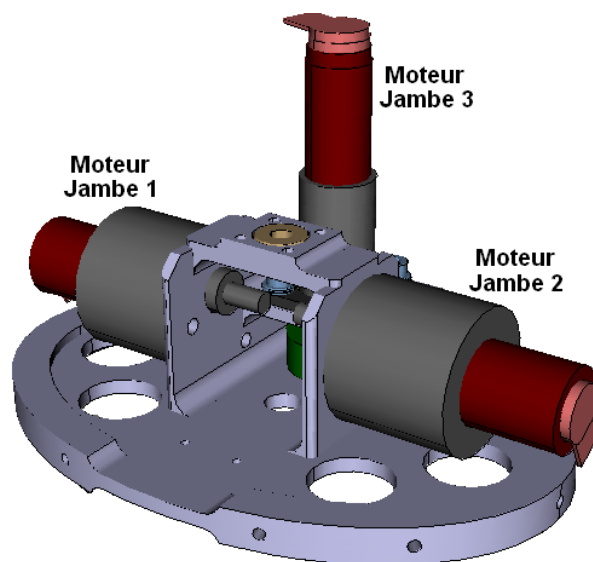


Figure B.1: – Placement des 3 moteurs sur un module

Les caractéristiques électriques et mécaniques de ces deux séries de moteurs sont donnés dans la documentation technique suivante :



DC-Micromotors

Graphite Commutation

5 mNm

For combination with (overview on page 14-15)

Gearheads:

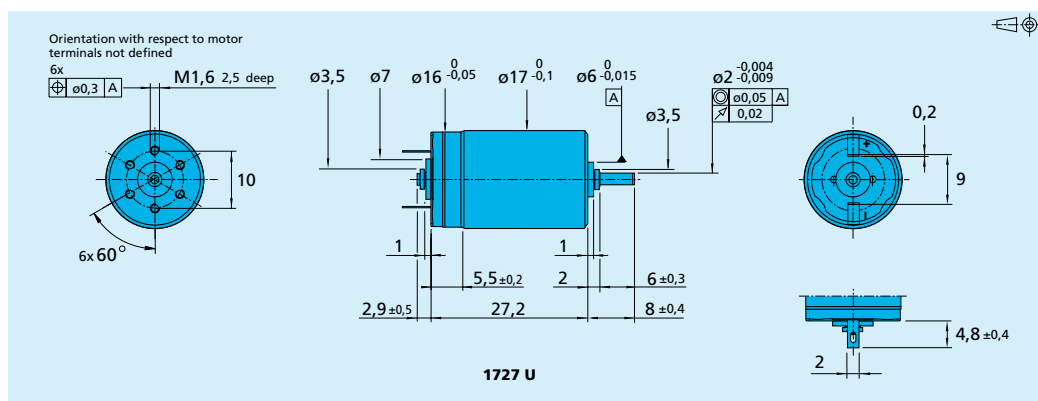
16/7, 20/1

Encoders:

IE2 – 16 ... 512

Series 1727 ... C

	1727 U	006 C	012 C	024 C	
1 Nominal voltage	U_N	6	12	24	Volt
2 Terminal resistance	R	3,0	13,8	57,6	Ω
3 Output power	$P_2 \text{ max.}$	2,37	2,25	2,25	W
4 Efficiency	$\eta \text{ max.}$	70	70	70	%
5 No-load speed	n_0	7 800	7 800	7 800	rpm
6 No-load current (with shaft \varnothing 2,0 mm)	I_0	0,055	0,026	0,013	A
7 Stall torque	M_H	11,6	11,0	11,0	mNm
8 Friction torque	M_R	0,36	0,35	0,36	mNm
9 Speed constant	k_n	1 460	700	343	rpm/V
10 Back-EMF constant	k_E	0,684	1,430	2,920	mV/rpm
11 Torque constant	k_M	6,53	13,60	27,90	mNm/A
12 Current constant	k_i	0,153	0,073	0,036	A/mNm
13 Slope of n-M curve	$\Delta n / \Delta M$	672	709	709	rpm/mNm
14 Rotor inductance	L	80	320	1 440	μH
15 Mechanical time constant	τ_m	9	9	9	ms
16 Rotor inertia	J	1,3	1,2	1,2	gcm^2
17 Angular acceleration	$\alpha \text{ max.}$	91	91	91	$\cdot 10^3 \text{ rad/s}^2$
18 Thermal resistance	R_{th1} / R_{th2}	5 / 24			K/W
19 Thermal time constant	τ_{w1} / τ_{w2}	4,2 / 254			s
20 Operating temperature range:					
– motor		– 30 ... + 100			$^{\circ}\text{C}$
– rotor, max. permissible		+ 125			$^{\circ}\text{C}$
21 Shaft bearings		ball bearings, preloaded			
22 Shaft load max.:					
– with shaft diameter		2,0			mm
– radial at 3 000 rpm (3 mm from bearing)		8			N
– axial at 3 000 rpm		0,8			N
– axial at standstill		10			N
23 Shaft play:					
– radial	\leq	0,015			mm
– axial	$=$	0			mm
24 Housing material		steel, black coated			
25 Weight		28			g
26 Direction of rotation		clockwise, viewed from the front face			
Recommended values - mathematically independent of each other					
27 Speed up to	$n_{e \text{ max.}}$	7 000	7 000	7 000	rpm
28 Torque up to	$M_{e \text{ max.}}$	5	5	5	mNm
29 Current up to (thermal limits)	$I_{e \text{ max.}}$	0,900	0,420	0,200	A



For details on technical information and lifetime performance refer to pages 28-34.
Edition 2006-2007

For options on DC-Micromotors refer to page 64.
Specifications subject to change without notice.
www.faulhaber-group.com



DC-Micromotors

Precious Metal Commutation

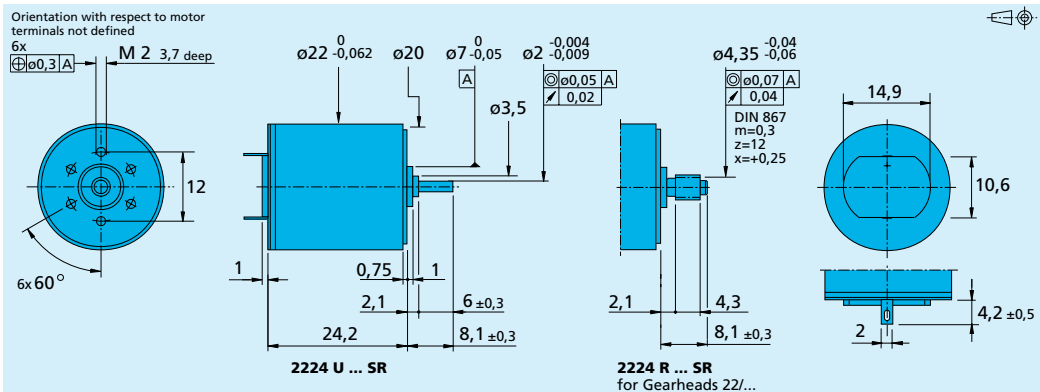
5 mNm

For combination with (overview on page 14-15)

Gearheads:
20/1, 22E, 22/2, 22/5, 22/6, 23/1, 38/3Encoders:
IE2 – 16 ... 512

Series 2224 ... SR

	2224 U	003 SR	006 SR	012 SR	018 SR	024 SR	036 SR	
1 Nominal voltage	U _N	3	6	12	18	24	36	Volt
2 Terminal resistance	R	0,56	1,94	8,71	17,50	36,30	91,40	Ω
3 Output power	P _{2 max.}	3,92	4,55	4,05	4,54	3,88	3,46	W
4 Efficiency	η _{max.}	80	82	82	82	81	80	%
5 No-load speed	n ₀	8 100	8 200	7 800	8 100	7 800	7 800	rpm
6 No-load current (with shaft ø 2,0 mm)	I ₀	0,066	0,029	0,014	0,010	0,007	0,005	A
7 Stall torque	M _H	18,5	21,2	19,8	21,4	19,0	16,9	mNm
8 Friction torque	M _R	0,23	0,2	0,2	0,21	0,2	0,22	mNm
9 Speed constant	k _n	2 730	1 380	657	454	328	219	rpm/V
10 Back-EMF constant	k _E	0,366	0,725	1,520	2,200	3,040	4,560	mV/rpm
11 Torque constant	k _M	3,49	6,92	14,50	21,00	29,10	43,50	mNm/A
12 Current constant	k _I	0,286	0,144	0,069	0,048	0,034	0,023	A/mNm
13 Slope of n-M curve	Δn/ΔM	438	387	394	379	411	462	rpm/mNm
14 Rotor inductance	L	11	45	200	450	800	1 800	μH
15 Mechanical time constant	τ _m	11	11	11	11	11	11	ms
16 Rotor inertia	J	2,4	2,7	2,7	2,8	2,6	2,3	gcm ²
17 Angular acceleration	α _{max.}	77	78	74	77	74	74	·10 ³ rad/s ²
18 Thermal resistance	R _{th 1} / R _{th 2}	5 / 20						K/W
19 Thermal time constant	τ _{w1} / τ _{w2}	6,8 / 440						s
20 Operating temperature range:								
– motor		– 30 ... + 85 (optional – 55 ... + 125)						°C
– rotor, max. permissible		+ 125						°C
21 Shaft bearings		sintered bronze sleeves		ball bearings		ball bearings, preloaded		
22 Shaft load max.:		(standard)		(optional)		(optional)		
– with shaft diameter		2,0		2,0		2,0		mm
– radial at 3 000 rpm (3 mm from bearing)		1,5		8		8		N
– axial at 3 000 rpm		0,2		0,8		0,8		N
– axial at standstill		20		10		10		N
23 Shaft play:								
– radial	≤	0,03		0,015		0,015		mm
– axial	≤	0,2		0,2		0		mm
24 Housing material		steel, black coated						
25 Weight		46						g
26 Direction of rotation		clockwise, viewed from the front face						
Recommended values - mathematically independent of each other								
27 Speed up to	n _{e max.}	8 000	8 000	8 000	8 000	8 000	8 000	rpm
28 Torque up to	M _{e max.}	5	5	5	5	5	5	mNm
29 Current up to (thermal limits)	I _{e max.}	2,200	1,200	0,570	0,400	0,280	0,180	A

For details on technical information and lifetime performance
refer to pages 28-34.
Edition 2006-2007For options on DC-Micromotors refer to page 64.
Specifications subject to change without notice.
www.faulhaber-group.com

Annexe C

Calcul du modèle dynamique direct d'un robot manipulateur sériel

Le modèle dynamique direct d'un robot série dont la base n'est pas fixe doit permettre de calculer les accélérations de la tête et articulaires en fonction des positions, vitesses et couples articulaires. L'algorithme est constitué de 3 récurrences basées sur les algorithmes de Newton-Euler et de Featherstone.

i) Récurrence avant :

Dans cette première étape, on calcule le torseur des vitesses des corps et les termes d'efforts qui ne dépendent pas des accélérations articulaires et de la tête pour $j=1, \dots, n$:

$${}^j\mathbb{T}_{j-1} = \begin{bmatrix} {}^j\mathbf{R}_{j-1} & -{}^j\mathbf{R}_{j-1} {}^{j-1}\hat{\mathbf{P}}_j \\ \mathbf{0}_{3 \times 3} & {}^j\mathbf{R}_{j-1} \end{bmatrix} \quad (\text{C.1})$$

$${}^j\mathbb{V}_j = {}^j\mathbb{T}_{j-1} {}^{j-1}\mathbb{V}_{j-1} + \dot{q}_j {}^j\mathbf{a}_j \quad (\text{C.2})$$

$${}^j\zeta_j = \begin{bmatrix} {}^j\mathbf{R}_{j-1} \left[{}^{j-1}\boldsymbol{\omega}_{j-1} \times ({}^{j-1}\boldsymbol{\omega}_{j-1} \times {}^{j-1}\mathbf{P}_j) \right] \\ {}^j\boldsymbol{\omega}_{j-1} \times \dot{q}_j {}^j\mathbf{a}_j \end{bmatrix} \quad (\text{C.3})$$

$${}^j\beta_j = {}^j\mathbf{f}_{ej} + \begin{bmatrix} {}^j\boldsymbol{\omega}_j \times ({}^j\boldsymbol{\omega}_j \times {}^j\mathbf{M}\mathbf{S}_j) \\ {}^j\boldsymbol{\omega}_j \times ({}^j\mathbf{J}_j {}^j\boldsymbol{\omega}_j) \end{bmatrix} \quad (\text{C.4})$$

ii) Récurrence arrière :

Dans cette étape, on cherche à déterminer l'accélération de la base du robot. Pour cela, on part de l'équation d'équilibre d'un corps :

$${}^j\mathbf{f}_j = {}^j\mathbb{F}_j + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} \quad (\text{C.5})$$

où :

$${}^j\mathbb{F}_j = {}^j\mathbb{J}_j {}^j\dot{\mathbb{V}}_j + {}^j\beta_j \quad (\text{C.6})$$

avec :

$${}^j\dot{\mathbf{V}}_j = {}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j\ddot{q}_j + {}^j\zeta_j \quad (\text{C.7})$$

ce qui donne :

$${}^j\mathbf{f}_j = {}^j\mathbb{J}_j \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j\ddot{q}_j + {}^j\zeta_j \right) + {}^{j+1}\mathbb{T}_j^T {}^{j+1}\mathbf{f}_{j+1} + {}^j\boldsymbol{\beta}_j \quad (\text{C.8})$$

Cependant, dans le cas du modèle direct, les accélérations articulaires ne sont pas connues. Pour les faire disparaître, on peut faire appel à l'équation du calcul des couples articulaires :

$$\boldsymbol{\tau}_j = {}^j\mathbf{a}_j^T {}^j\mathbf{f}_j + \mathbf{I}a_j\ddot{q}_j \quad (\text{C.9})$$

où

$$\boldsymbol{\tau}_j = \Gamma_j - F_{sj}\text{sign}(\dot{q}_j) - F_{vj}\dot{q}_j \quad (\text{C.10})$$

Pour initialiser l'algorithme, il faut remarquer que pour le corps n ${}^{n+1}\mathbf{f}_{n+1} = 0$ (le corps n représente l'organe terminal). En utilisant les équations (C.8) et (C.9) pour $j=n$, on obtient dans un premier temps :

$${}^n\mathbf{f}_n = {}^n\mathbb{J}_n \left({}^n\dot{\mathbf{V}}_{n-1} + {}^n\mathbf{a}_n\ddot{q}_n + {}^n\zeta_n \right) + {}^n\boldsymbol{\beta}_n \quad (\text{C.11})$$

Puis, en introduisant (C.11) dans (C.9), cela nous permet de calculer \ddot{q}_n en fonction de ${}^n\dot{\mathbf{V}}_{n-1}$:

$$\ddot{q}_n = \mathbf{H}_n^{-1} \left(-{}^n\mathbf{a}_n^T {}^n\mathbb{J}_n \left({}^n\dot{\mathbf{V}}_{n-1} + {}^n\zeta_n \right) + \boldsymbol{\tau}_n - {}^n\mathbf{a}_n^T {}^n\boldsymbol{\beta}_n \right) \quad (\text{C.12})$$

Finalement, on peut réécrire l'équation d'équilibre en introduisant (C.12) dans (C.11) :

$${}^n\mathbf{f}_n = {}^n\mathbb{K}_n {}^n\dot{\mathbf{V}}_{n-1} + {}^n\boldsymbol{\alpha}_n \quad (\text{C.13})$$

avec :

$$\mathbf{H}_n = {}^n\mathbf{a}_n^T {}^n\mathbb{J}_n {}^n\mathbf{a}_n + \mathbf{I}a_n \quad (\text{C.14})$$

$${}^n\mathbb{K}_n = {}^n\mathbb{J}_n - {}^n\mathbb{J}_n {}^n\mathbf{a}_n \mathbf{H}_n^{-1} {}^n\mathbf{a}_n^T {}^n\mathbb{J}_n \quad (\text{C.15})$$

$${}^n\boldsymbol{\alpha}_n = {}^n\mathbb{K}_n {}^n\zeta_n + {}^n\mathbb{J}_n {}^n\mathbf{a}_n \mathbf{H}_n^{-1} (\boldsymbol{\tau}_n - {}^n\mathbf{a}_n^T {}^n\boldsymbol{\beta}_n) + {}^n\boldsymbol{\beta}_n \quad (\text{C.16})$$

-Calcul pour le corps n-1 :

Pour le corps n-1, l'équation d'équilibre (C.5) (après avoir remplacé j par n-1) se retrouve sous sa forme générale suivante :

$${}^{n-1}\mathbb{J}_{n-1} {}^{n-1}\dot{\mathbf{V}}_{n-1} = {}^{n-1}\mathbf{f}_{n-1} - {}^n\mathbb{T}_{n-1}^T {}^n\mathbf{f}_n - {}^{n-1}\boldsymbol{\beta}_{n-1} \quad (\text{C.17})$$

En substituant ${}^n\mathbf{f}_n$ dans (C.13) par l'expression simplifiée que l'on a trouvée précédemment (équation (C.13)), on trouve :

$${}^{n-1}\mathbf{f}_{n-1} = {}^{n-1}\mathbb{J}_{n-1}^* \left({}^{n-1}\dot{\mathbf{V}}_{n-2} + {}^{n-1}\mathbf{a}_{n-1}\ddot{\mathbf{q}}_{n-1} + {}^{n-1}\boldsymbol{\zeta}_{n-1} \right) + {}^{n-1}\boldsymbol{\beta}_{n-1}^* \quad (\text{C.18})$$

avec :

$${}^{n-1}\mathbb{J}_{n-1}^* = {}^{n-1}\mathbb{J}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\mathbb{K}_n {}^n\mathbb{T}_{n-1} \quad (\text{C.19})$$

$${}^{n-1}\boldsymbol{\beta}_{n-1}^* = {}^{n-1}\boldsymbol{\beta}_{n-1} + {}^n\mathbb{T}_{n-1}^T {}^n\boldsymbol{\alpha}_n \quad (\text{C.20})$$

En remarquant que l'équation (C.18) possède la même forme que l'équation (C.11), ainsi on déduit :

$${}^{n-1}\mathbf{f}_{n-1} = {}^{n-1}\mathbb{K}_{n-1} {}^{n-1}\dot{\mathbf{V}}_{n-2} + {}^{n-1}\boldsymbol{\alpha}_{n-1} \quad (\text{C.21})$$

En généralisant cette approche pour les autres corps (j=n,...,1) :

$${}^j\mathbf{f}_j = {}^j\mathbb{K}_j {}^j\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\alpha}_j \quad (\text{C.22})$$

$$\ddot{\mathbf{q}}_j = \mathbf{H}_j^{-1} \left(-{}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\zeta}_j \right) + \boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^* \right) \quad (\text{C.23})$$

avec :

$$\mathbf{H}_j = {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j + \mathbf{I}_{a_j} \quad (\text{C.24})$$

$${}^j\mathbb{K}_j = {}^j\mathbb{J}_j^* - {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j \mathbf{H}_j^{-1} {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \quad (\text{C.25})$$

$$\boldsymbol{\tau}_j = \boldsymbol{\Gamma}_j - \mathbf{F}_{sj} \text{sign}(\dot{\mathbf{q}}_j) - \mathbf{F}_{vj} \dot{\mathbf{q}}_j \quad (\text{C.26})$$

$${}^j\boldsymbol{\alpha}_j = {}^j\mathbb{K}_j {}^j\boldsymbol{\zeta}_j + {}^j\mathbb{J}_j^* {}^j\mathbf{a}_j \mathbf{H}_j^{-1} (\boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^*) + {}^j\boldsymbol{\beta}_j^* \quad (\text{C.27})$$

$${}^{j-1}\mathbb{J}_{j-1}^* = {}^{j-1}\mathbb{J}_{j-1} + {}^j\mathbb{T}_{j-1}^T {}^j\mathbb{K}_j {}^j\mathbb{T}_{j-1} \quad (\text{C.28})$$

$${}^{j-1}\boldsymbol{\beta}_{j-1}^* = {}^{j-1}\boldsymbol{\beta}_{j-1} + {}^j\mathbb{T}_{j-1}^T {}^j\boldsymbol{\alpha}_j \quad (\text{C.29})$$

On note que pour la tête ($j=0$), les équations (C.24)...(C.27) ne s'appliquent pas du fait qu'il n'existe pas de moteur. En utilisant l'équation (C.5) et en remarquant que ${}^0\mathbf{f}_0$ est égal à zéro, on obtient

$$0 = {}^0\mathbb{J}_0 {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0 + {}^1\mathbf{T}_0^T {}^1\mathbf{f}_1 \quad (\text{C.30})$$

En utilisant l'équation (C.22) pour $j=1$, l'équation (C.30) devient alors :

$$0 = {}^0\mathbb{J}_0 {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0 + {}^1\mathbf{T}_0^T \left({}^1\mathbb{K}_1 {}^1\mathbf{T}_0 {}^0\dot{\mathbf{V}}_0 + {}^1\boldsymbol{\alpha}_1 \right) \quad (\text{C.31})$$

$$0 = {}^0\mathbb{J}_0^* {}^0\dot{\mathbf{V}}_0 + {}^0\boldsymbol{\beta}_0^* \quad (\text{C.32})$$

La résolution de cette équation donne :

$${}^0\dot{\mathbf{V}}_0 = - \left({}^0\mathbb{J}_0^* \right)^{-1} {}^0\boldsymbol{\beta}_0^* \quad (\text{C.33})$$

Pour résumer, on commence par initialiser, ${}^n\mathbb{J}_n^* = {}^n\mathbb{J}_n$, ${}^n\boldsymbol{\beta}_n^* = {}^n\boldsymbol{\beta}_n$ puis on calcule pour $j=n, \dots, 1$ les équations (C.24) à (C.29). Une fois terminée, l'accélération de la tête de l'anguille ${}^0\dot{\mathbf{V}}_0$ est déduite en utilisant l'équation (C.33).

iii) Récurrence avant :

A présent, on peut calculer le torseur ${}^j\mathbf{f}_j$ (si nécessaire) ainsi que les accélérations articulaires $\ddot{\mathbf{q}}_j$ pour $j=1, \dots, n$ en utilisant les équations (C.22) et (C.23) :

$$\ddot{\mathbf{q}}_j = \mathbf{H}_j^{-1} \left[- {}^j\mathbf{a}_j^T {}^j\mathbb{J}_j^* \left({}^j\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\zeta}_j \right) + \boldsymbol{\tau}_j - {}^j\mathbf{a}_j^T {}^j\boldsymbol{\beta}_j^* \right] \quad (\text{C.34})$$

$${}^j\mathbf{f}_j = {}^j\mathbb{K}_j {}^j\mathbf{T}_{j-1} {}^{j-1}\dot{\mathbf{V}}_{j-1} + {}^j\boldsymbol{\alpha}_j \quad (\text{C.35})$$

où :

$${}^j\dot{\mathbf{V}}_j = {}^j\dot{\mathbf{V}}_{j-1} + {}^j\mathbf{a}_j \ddot{\mathbf{q}}_j + {}^j\boldsymbol{\zeta}_j \quad (\text{C.36})$$

Annexe D

Calcul des modèles dynamiques des jambes d'un module

Pour le calcul de \mathbf{H}_i représentant le modèle dynamique inverse de la jambe i , on utilise le logiciel de calcul symbolique SYMORO+ [Khalil et Creusot, 1997]. On définit les paramètres inertiels d'un corps k et le vecteur gravité tel que :

$$\mathbf{I}_k = \begin{bmatrix} XXk & XYk & XZk \\ XYk & YYk & YZk \\ XZk & YZk & ZZk \end{bmatrix}, \quad \mathbf{MS}_k = [MXk \quad MYk \quad MZk], \quad \mathbf{G} = [G1 \quad G2 \quad G3]$$

La figure (D.1) représente la numérotation utilisée pour les paramètres inertiels des corps :

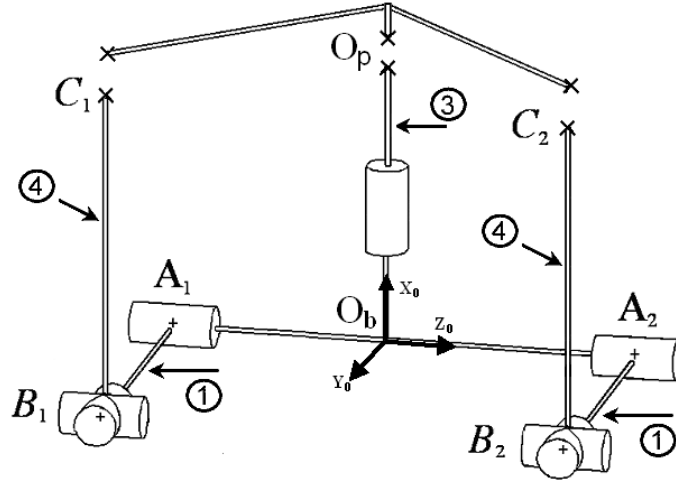


Figure D.1: – Numérotation des corps pour les paramètres inertiels

Pour la jambe 3, le modèle dynamique s'écrit :

$$H3 = ZZ3*QDP3 + IA3*QDP3 + FV3*QP3 + FS3*sign(QP3);$$

Pour les 2 autres jambes, on définit $e_i=1$ pour la jambe 1 et $e_i=-1$ pour la jambe 2. Le modèle dynamique s'écrit alors :

```

S1=sin(t1);  C1=cos(t1);
S2=sin(t2);  C2=cos(t2);
S3=sin(t3);  C3=cos(t3);
DV331=-QP1.^2;
VP11=-(C1.*G1) - ei*G2.*S1;
VP21=-ei*(C1.*G2) + G1.*S1;
No31=QDP1.*ZZ1;
W32=QP1 + QP2;
WP32=QDP1 + QDP2;
VSP12=D2.*DV331 + VP11;
VSP22=D2.*QDP1 + VP21;
VP12=C2.*VSP12 + S2.*VSP22;
VP22=-(S2.*VSP12) + C2.*VSP22;
WI13=-ei*(S3.*W32);
WI23=-ei*(C3.*W32);
WP13=QP3.*WI23 - ei*S3.*WP32;
WP23=-(QP3.*WI13) - ei*C3.*WP32;
DV223=-WI23.^2;
DV333=-QP3.^2;
DV123=WI13.*WI23;
DV133=QP3.*WI13;
U113=DV223 + DV333;
U213=DV123 + QDP3;
U313=DV133 - WP23;
VP13=G3.*S3 + C3.*VP12;
VP23=C3.*G3 - S3.*VP12;
DV114=-WI13.^2;
DV224=-QP3.^2;
DV334=-WI23.^2;
DV124=QP3.*WI13;
DV134=-(WI13.*WI23);
DV234=-(QP3.*WI23);
U114=DV224 + DV334;
U124=DV124 + WP23;
U134=DV134 + QDP3;
U214=DV124 - WP23;
U224=DV114 + DV334;
U234=DV234 - WP13;
U314=DV134 - QDP3;
U324=DV234 + WP13;
U334=DV114 + DV224;
VSP14=D4.*U113 + VP13;
VSP24=D4.*U213 + VP23;
VSP34=D4.*U313 + ei*VP22;
F14=MX4.*U114 + MY4.*U124 + MZ4.*U134 + M4.*VSP14;
F24=MX4.*U214 + MY4.*U224 + MZ4.*U234 + M4.*VSP34;
F34=MX4.*U314 + MY4.*U324 + MZ4.*U334 - M4.*VSP24;
PIS14=-YY4 + ZZ4;
PIS24=XX4 - ZZ4;

```



```

PIS34=-XX4 + YY4;
No14=DV234.*PIS14 + WP13.*XX4 - U314.*XY4 + U214.*XZ4 + (-DV224 + DV334).*YZ4;
No24=DV134.*PIS24 + U324.*XY4 + (DV114 - DV334).*XZ4 + QDP3.*YY4 - U124.*YZ4;
No34=DV124.*PIS34 + (-DV114 + DV224).*XY4 - U234.*XZ4 + U134.*YZ4 - WP23.*ZZ4;
N14=No14 - MY4.*VSP24 - MZ4.*VSP34;
N24=No24 + MZ4.*VSP14 + MX4.*VSP24;
N34=No34 - MY4.*VSP14 + MX4.*VSP34;
N23=-(D4.*F24) - N34;
N33=-(D4.*F34) + N24;
FDI13=C3.*F14 + F34.*S3;
N32=-ei*(C3.*N23) - ei*N14.*S3;
FDI22=ei*(C2.*F24) + FDI13.*S2;
N31=D2.*FDI22 + N32 + No31 - MY1.*VP11 + MX1.*VP21;
GAM1=N31 + IA1.*QDP1 + FV1.*QP1 + FS1.*sign(QP1);
GAM2=N32;
GAM3=N33;
Hi = [GAM1;GAM2;GAM3];

```

Pour pouvoir calculer les positions et vitesses articulaires (Q_k , QPk) des différentes liaisons de la jambe i , on utilise tout d'abord le calcul du modèle géométrique :

```

Y = -Pz - ei*R1;
SQ = Y/D4;
t3 = atan2( SQ , sqrt(1 - SQ^2) );
W=-(D4*cos(t3));
B1 = -2*ei*D2*Py;
B2 = -2*D2*Px;
B3 = -D2^2 - Px^2 - Py^2 + W^2;
SQ = (B1*B3 - ei*B2*sqrt(B1^2 + B2^2 - B3^2))/(B1^2 + B2^2);
CQ = (B2*B3 + ei*B1*sqrt(B1^2 + B2^2 - B3^2))/(B1^2 + B2^2);
t1 = atan2( SQ , CQ );
t2 = atan2((-ei*Py*cos(t1) + Px*sin(t1))/W , (D2 - Px*cos(t1) - ei*Py*sin(t1))/W );
q1 = t1;
q2 = t2;
q3 = t3;

```

Puis, on calcule la matrice jacobienne en utilisant les équations qui suivent :

```

INVJ(1,1) = (cos(t1 + t2)*Csc(t2))/D2;
INVJ(2,1) = -((Csc(t2)*(D4*cos(t1 + t2) + D2*cos(t1)*Sec(t3)))/(D2*D4));
INVJ(3,1) = 0;
INVJ(1,2) = ei*(Csc(t2)*sin(t1 + t2))/D2;
INVJ(2,2) = -ei*Csc(t2)*((Sec(t3)*sin(t1))/D4 - ei*sin(t1 + t2)/D2);
INVJ(3,2) = 0;
INVJ(1,3) = -((Csc(t2)*tan(t3))/D2);
INVJ(2,3) = (Csc(t2)/D2 + (Cot(t2)*Sec(t3))/D4)*tan(t3);
INVJ(3,3) = -(Sec(t3)/D4);

```

Pour finir, l'obtention des accélérations articulaires $QDPk$ nécessite le calcul du vecteur $\ddot{\mathbf{J}}\mathbf{q}$. Pour cela, on utilise les équations ci-après :

```

S2=sin(t2);
C2=cos(t2);
S3=sin(t3);
C3=cos(t3);
DV331=-QP1^2;
W32=QP1 + QP2;
VSP12=D2*DV331;
VPJ12=C2*VSP12;
VPJ22=-(S2*VSP12);
WI13=-ei*(S3*W32);
WI23=-ei*(C3*W32);
WPJ23=-(QP3*WI13);
DV223=-WI23^2;
DV333=-QP3^2;
DV123=WI13*WI23;
DV133=QP3*WI13;
U113=DV223 + DV333;
U313=DV133 - WPJ23;
VPJ13=C3*VPJ12;
VPJ23=-(S3*VPJ12);
VSP14=D4*U113 + VPJ13;
VSP24=D4*DV123 + VPJ23;
VSP34=D4*U313 + ei*VPJ22;
VPJ14=VSP14;
VPJ24=VSP34;
VPJ34=-VSP24;

JPQP(1,1) = VPJ14;
JPQP(2,1) = VPJ24;
JPQP(3,1) = VPJ34;

```


MODELISATION DYNAMIQUE ET COMMANDE D'UN ROBOT ANGUILE

Résumé : Cette thèse se place dans le cadre du projet Robot-Anguille, qui regroupe six laboratoires français. L'objectif du projet est de concevoir, étudier et réaliser un robot "anguille" ou "serpent-nageur" capable de se déplacer dans un espace à 3 dimensions.

A travers cet objectif, la thèse se porte dans un premier temps sur la modélisation dynamique du robot sous la forme d'un mécanisme hybride (structures de robots parallèles montées en série) permettant ainsi d'être le plus proche possible du prototype construit. Pour cela nous avons utilisé les algorithmes récursifs de Newton-Euler pour les modèles dynamiques inverse et direct en les généralisant au cas des robots à base mobile. Nous avons également proposé un modèle de contact fluide-structure pour simuler le comportement du robot dans l'eau. Pour tester ces algorithmes, nous avons simulé le comportement du robot lors de différents types de nage et en avons tiré des conclusions qui nous ont guidées dans la conception du prototype.

Dans un deuxième temps, à partir d'un générateur de mouvements à base de CPGs (ou réseau de neurones), nous avons étudié des lois de commande pour réaliser des simulations de nage en boucle fermée. Ainsi, nous avons abordé les problèmes de la nage vers des points cible et l'évitement d'obstacles pour la nage en milieu confiné.

Mots-clés : Modélisation dynamique, Robots nageurs , Robot anguille , Réseaux de neurones.

DYNAMIC MODELING AND CONTROL OF AN EEL LIKE ROBOT

Abstract : This thesis takes place in the framework of the project "Robot-Anguille", which gathers six French laboratories. The objective of the project is to design and produce an "eel-like" robot or "snake-swimmer" able to swim in a 3 dimensional space.

Through this guideline, the thesis treats at first the dynamic modeling of the robot of a hybrid mechanism structure(parallel robots connected in serie) which correspond to the prototype built. To achieve that task, we have used recursive Newton-Euler algorithms for the inverse and direct dynamic models by generalizing them for robots with a mobile base. We also proposed a fluid-structure interaction model to simulate the behaviour of the robot in water. To test these algorithms, we have simulated the behaviour of the robot for various types of swimming and drew some conclusions which have guided us on the design of the prototype.

Secondly, starting from a motion law based on CPGs (or network of neurons), we studied closed loop control laws to carry out swimming simulations. Thus, we tackled the problem of swimming towards target points and the problem of obstacle avoidance in confined surroundings.

Keywords : Dynamic Modeling, Swimming structures , Eel like robot , Neuron network.

Discipline : Sciences de l'Ingénieur